Mathematics Clinic

Mathematics Clinic Handbook

2019

2019 Edition; Revision 1.0. (April 30, 2019)

Calendar

Note that dates and deadlines are subject to change.

Date	Description	Page
September 4 (Tue)	Student Orientation Day (4:15 P.M.–5:30 P.M.; Sprague 3)	5
September 4 (Tue)	Project Managers Meeting (5:30 P.M.–6:30 P.M. in Sprague 3)	
September 6–18	Marathon Push (Including Preparation of Your Statement of Work)	8
September 6 (Thu)	Clinic/Sponsor Orientation Day (11:00 A.M.–1:00 P.M.; Sprague 3 workrooms) *Be prepared to ask your liaisons questions.*	6
September 7 (Fri)	All Forms Due by 4:00 P.M. to Molly Reeves (SCTL 2404)	5
September	Site Visits with Sponsor (Arranged by Teams and Sponsors)	9
September 11 (Tue)	Professional Development: Teleconference and Site Visit Etiquette (11:00–12:15 P.M.; Galileo McAlister; Combined with CS)	10
September 18 (Tue)	Statement of Work Peer Review (11:00–12:15 P.M.; Sprague 3) Bring Advisor-Previewed Draft.	8
September 18 (Tue)	Project Managers Meeting (11:00–12:15 P.M.; Sprague 3)	
September 20 (Thu)	Engineering Career Fair	
September 21 (Fri)	Peer Reviews of Statement of Work Due to Team	8
September 24 (Mon)	Advisor-Previewed Project Budget Due to Clinic Director	53
September 25 (Tue)	Review Statement of Work According to Slides on Project Management & Planning Sent by Weiqing Gu (11:00–12:15 P.M.; Sprague 3)	10
September 28 (Fri)	Advisor-Previewed Statement of Work Due to Clinic Director	8
October	Statement of Work Negotiations and Approval from Liaison	8
October 9 (Tue)	Project Managers Meeting (11:00 A.M.–12:15 P.M.; Sprague 3)	
October 11 (Thu)	STEM Career Fair	
October 22–23	Fall Break	

continued on following page

Date	Description	Page
October 30 (Tue)	Learn About and Discuss Effective Presentations & Posters Using Slides Sent by Weiqing Gu	10
November 13 (Tue)	Project Managers Meeting (11:00–12:15 P.M.; Sprague 3)	
November 20 (Tue)	Informal Practice Presentations and Lunch (11:00 A.M.–1:00 P.M.; Sprague 3)	9
November 21–23	Thanksgiving Break	
November 28 (Wed)	Advisor-Previewed Midyear Report Draft Due to Clinic Director	10
December 14 (Fri)	Final Version of Midyear Report Due to Clinic Director and Coordinator (by 4:00 P.M.)	10
December 14 (Fri)	Individual and Team Self-Evaluation Due to Clinic Coordinator (by 4:00 P.M.)	13
December 22–January 20	Winter Break	
January 24–January 28	Workrooms Used for Mathematics Contest in Modeling	10
January 29 (Tentative)	Formal Presentations for Spring Semester Begin (All Clinics—**Attendance Is Mandatory**)	9
TBD	Spring Career Fair	
March 18–22	Spring Break	
April	Site Visits with Sponsor (Arranged by Teams and Sponsors)	9
April 5 (Fri)	Advisor-Previewed Poster Draft Due to Clinic Director and Clinic Coordinator by 4:00 P.M.	11
April 15 (Mon)	Draft of Final Report Due to Advisor	11
April 16 (Tue)	Liaison-Approved Final Poster Completed and Submitted for Printing to Molly Reeves and Dru-Ann Thomas by 4:00 P.M.	11
April 24 (Wed)	Advisor-Previewed Draft of Final Report Due to Clinic Director	11
April 26 (Fri)	Individual and Team Self-Evaluation Due to Clinic Coordinator and cc'd to Clinic Director (by 4:00 PM)	13
May 3 (Fri)	Poster session on Alumni Weekend (1:30 P.M.–2:30 P.M.; Shanahan 3481) Posters need to be hung by 1:20 P.M.	11
May 7 (Tue)	Projects Day and Formal Dinner with Liaisons	11

continued on following page

Date	Description	Page
May 10 (Fri)	Final Report Due to Advisor and Clinic Director	11, 94
May 10 (Fri)	Printed Final Reports and CD-ROMs Turned In and Closing Checklist Completed and Delivered to DruAnn Thomas; E-Mail PDF of Report to DruAnn Thomas (by 4:00 P.M.)	93, 94, 94, 95, 98
May 10 (Fri)	Final Work Products Archived Electronically (Report to Molly Reeves)	11, 95, 97
May 10 (Fri)	Senior Dinner (Clinic & Thesis)	

May 10 (Fri) - Final Report due; Archive and submit all work products

Deliver printed Final Report and a hard-drive with all the clinic team's work (just zip all clinic related files and save onto the hard drive) including data, weekly conference call slides, meeting notes if any, code, other methods tried, all .tex files with figures, references etc. that generate the .pdf files for midterm, final reports, conference slides, and posters. (Note: the other methods are those the students tried but may not work, however, not working results are also results since the next clinic team does not have to retry it again.) The team will also prepare a thumb drive for the final working results and corresponding code, final report and poster including the .tex files for liaison(s) together with a read-me or a diagram to explain the files included.

Contact Information

Computing Help

Contact the department's information technology analyst at `mreeves@hmc.edu`.

People

Name	Office	Extension	E-Mail Address
Weiqing Gu, Director	SCTL 3420	18929/18570	gu@math.hmc.edu
DruAnn Thomas, Clinic Coordinator	Olin 1257	18570	druann_thomas@hmc.edu
Lisette de Pillis, Chair	SCTL 3410	18975/18023	depillis@math.hmc.edu
Jocelyn Olds-McSpadden, Math Administrative Aide	SCTL 3428	18023	jocelyn@math.hmc.edu
Molly Reeves, Information Technology Analyst	SCTL 2404	18754	mreeves@hmc.edu
A/V Support (CIS)	Sprague 5	77777	helpdesk@hmc.edu

Facilities

Name	Room Number	Notes
Sprague Third Floor[†]	π (east)	Conference phone: (909) 607-4665
	e (west)	Conference phone: (909) 607-4685
	Clinic 1	Computers; conference phone: (909) 607-3219
	Clinic 2	Computers; conference phone: (909) 607-8419
	Clinic 3	Computers; conference phone: (909) 607-8409
Math Workroom	SCTL 3433	Networked copier; color laser printer; fax machine: (909) 621-8366

[†] Requires card-key access through F&M; see DruAnn Thomas for access.

Websites

Website	URL
Mathematics Clinic	https://www.math.hmc.edu/clinic/
Clinic Participants	https://www.math.hmc.edu/clinic/participants/
Clinic Forms	https://www.math.hmc.edu/clinic/participants/forms/
Math Computing	https://www.math.hmc.edu/computing/
Clinic LaTeX Class	https://www.math.hmc.edu/computing/support/tex/classes/hmcclinic/
`hmcmath` BibTeX Style	https://www.math.hmc.edu/computing/support/tex/bibliography-styles/
Sample Clinic Report	https://www.math.hmc.edu/computing/support/tex/sample-report/
Clinic Calendar on Google	https://www.math.hmc.edu/clinic/participants/calendar/

Contents

List of Figures

List of Tables

Acknowledgments

With major input and appendices from Harvey Mudd College's Mathematics Clinic Faculty Advisors Committee and the Departments of Biology, Chemistry, Computer Science, Engineering, Humanities, and Physics, and the Environmental Studies Program.

Substantial contributions to this document were made by (in alphabetical order) Andrew Bernoff, Claire Connelly, Lisette de Pillis, Weiqing Gu, Kerry Karukstis, Rachel Levy, Susan Martonosi, Melissa O'Neill, Michael Raugh, Barbara Schade, DruAnn Thomas, Lesley Ward.

Claire Connelly undertook the reorganization and rewriting of this document during the summers of 2006 and 2007, based on the last major revision written and assembled by Michael Raugh. A second significant revision was made during the summer of 2008 with a third revision during the summer of 2010.

Another significant revision due to a switch to the use of Git as the recommended version-control system took place in 2015.

Chapter 1

The Clinic Program

The Harvey Mudd College community places a high value on developing research skills in its students that they can carry into their future careers, in academia, industry, or elsewhere.

The Department of Mathematics has two options for students to choose from as their "capstone research experience": senior thesis and Clinic.

Both programs allow you to experience the rigors of scientific and mathematical research—constructing a unique research problem; combing the literature for relevant work; clearly stating one or more hypotheses and developing proofs or computational programs to test those hypotheses; and, finally, writing up the results in a professional form.

1.1 Teamwork

Senior-thesis problems are worked on by a single researcher, working largely alone. They are also more likely to feature "pure" research problems, often with less relevance to real-world purposes than Clinic projects.

In contrast, the Mathematics Clinic program provides you with the opportunity to work on real-world mathematical problems as part of a team of researchers. The team approach is used for two reasons. First, the problems accepted for Clinic projects are typically too complex for a single person to solve within the span of a single academic year. Second, and perhaps more importantly, the team approach gives students the chance to experience the sorts of team-based research that is common in graduate school, industry, and academic life.

A Clinic team typically consists of four students, one faculty advisor,

and one or more liaisons appointed by your sponsor to monitor your team's progress and serve as a conduit for obtaining resources (information, equipment, advice) from your project's sponsor. Some projects may also engage consultants to provide specialized assistance.

You and your teammates will develop skills in cooperation and consensus building. You will learn how to realistically assess your skills and those of your teammates; weigh your skills against the problem; work out a division of labor; and write a well-informed Statement of Work spelling out what you believe your team can reasonably do in the available time and with the available resources.

You will learn how to negotiate the terms of the Statement of Work with your sponsor's liaisons; later, as your understanding of the problem evolves, you may decide to renegotiate the terms as part of a mid-course correction.

Along the way you will discover the importance of project-management tools and techniques in efficiently coordinating work and helping you stay on schedule. You will create technical presentations and practice giving them to audiences of varying sizes and makeups, and you will write a professional-quality research report.

These skills are all valuable in themselves, but perhaps more important than the skills you will learn is the discovery that you can achieve far more than you thought possible. The self-confidence you will develop will join with the skills you learn to serve you well in many walks of life.

1.1.1 Project Management

Your team will select one or more members to act as project managers during the course of the project.[1] Clinic projects afford a rare opportunity for an undergraduate to directly experience the management role in research and to work with their peers in planning work and performing it in an organized way.

The project manager works to promote team consensus on goals and objectives; coordinates and enforces individual responsibility for specific tasks and division of labor; keeps the team on schedule; and is the key communicator between the members of your team, your faculty advisor, and your sponsor's liaison(s).

[1] Some projects may have more than one project manager, with managers working together or with one student managing the project one semester and a second taking over for the second semester.

Your faculty advisor and the Clinic director act as advisors to your project manager and your team. Although your team's advisor takes some responsibility for the final state of your reports and other deliverables, your team's student project manager is in charge of your team, and your project manager and your team are primarily responsible for all work on the project.

More details about the skill requirements and responsibilities of the project-manager position are available in Appendix F.

1.1.2 Your Faculty Advisor

Your faculty advisor serves in a role somewhat like a consulting expert who, as a professional mathematician, provides guidance on the general approach to doing research (such as in the performance of the preliminary literature search and ongoing background study); suggestions about possible approaches to a solution; and occasional discussion or lectures about fine points of your evolving analysis in areas where your advisor's knowledge may be relevant. Your advisor also serves as a coach and quality-control system for the structuring and preparation of your team's oral presentations and written reports and other documentation.

1.1.3 The Clinic Director

The mathematics Clinic director takes a higher-level coordinating role in managing the Clinic program and the various departmental Clinic projects. Among other tasks, the Clinic director works with sponsors to find and select potential Clinic projects; helps conduct the business and administrative side of getting projects under way; acts as an advisor to the faculty advisors; works with the college's other Clinic directors to arrange and coordinate events such as Projects Day; and acts as an ultimate resource for your team. If there are conflicts between your team's view of the project and the sponsor's view, or if legal or ethical questions arise during the course of your project, the Clinic director will help you resolve those questions.

1.1.4 Your Sponsor's Liaison

Your sponsor's liaison is the connecting bridge between the sponsoring organization and your team, offering domain expertise on the problem and ensuring that your team's research satisfies the sponsor's goals and adheres to the sponsor's constraints and policies.

Your liaison's mathematical background may range from none at all to decades of research experience, but they will be experts in the application area of the problem you are trying to solve and are thus a valuable resource for your team, with several hours a week dedicated to working with you on your project.

You should ensure that your liaison is well-informed about your progress throughout the year, and also turn to your liaison when questions arise about the direction of the project.

Keep in mind that your liaison *must approve the content of any public presentations* about your project, so you will need to prepare your slides and posters well in advance of your presentation to allow time to make any necessary changes.

1.2 Clinic Problems

Clinic problems come to us from organizational sponsors—companies, not-for-profit research laboratories, government agencies, even Harvey Mudd College or one of the other Claremont Colleges. These problems are seen by their proposers as important and worthy of investigation, but they may be too vaguely defined to provide a clear research program. Thus your team must begin by refining your sponsor's project proposal into a form that can be understood, analyzed, and solved within the academic year the team is assembled. This refinement and analysis approach to problem solving is common to both pure and applied mathematical research.

1.3 History

The history of the Mathematics Clinic program is described in greater detail in Appendix J.

Chapter 2

Project Events and Milestones

Every project shares a basic set of events and milestones. Some of these events, such as deadlines for submission of various deliverables, or shared events such as Projects Day take place on particular days, whereas others, such as site visits and presentations, may vary considerably from project to project based on choices made by the team members and their advisor in consultation with their project's liaison.

A Google calendar with the key shared dates is viewable on the Clinic website, at https://www.math.hmc.edu/clinic/participants/calendar/. It includes a link allowing you to subscribe to the calendar.

This chapter is meant to provide a quick overview of the common events and milestones. Additional information about participating in these events or meeting deadlines may be provided in subsequent chapters, appendices, or handouts, and will be noted in the appropriate discussions.

2.1 Orientation

There are two major events associated with the startup process, *Student Orientation Day* and *Sponsor Orientation Day*.

Student Orientation Day

On Student Orientation Day, September 4 (Tue), you will be introduced to the projects and their faculty advisors. You will meet as a team, and deal with some of the administrative tasks required by your participation in the Clinic program.

In particular, you will be asked to fill in and sign four major forms for the department:

1. *Mathematics Clinic Project Participant Confidentiality Acknowledgment*

   This document is a legal contract in which you acknowledge the sponsor's ownership of the work products produced by you and your team during your participation in the project and agree to not discuss details of the project with people outside of your team.

2. *Information and Expectations for the Senior Research Experience*

   This form spells out some basic commitments that you are making by agreeing to take part in a Clinic project, such as attending regular meetings, diligently pursuing the research goals of the project, producing professional reports, and so forth. (This form is also signed by thesis students.)

3. *Mathematics Account Request Form*

   This form is an agreement between you and the Mathematics Department to adhere to the Department's Appropriate-Use Policy for Computing Resources. A copy of this policy can be found at https://www.math.hmc.edu/computing/policies/appropriate-use/appropriate-use.pdf.

4. *Authorization to Obtain Motor Vehicle Record*

   This form authorizes Harvey Mudd College to obtain your driving record and, based on it, determine your eligibility to drive College-owned vehicles for College purposes such as Clinic. It also details some of the policies concerning use of College-owned vehicles. Clinic teams are strongly advised to use College-owned vehicles for any local Clinic-related travel, such as local site visits or trips to the airport.

These forms are due to DruAnn Thomas by **September 7 (Fri)**.

Clinic/Sponsor Orientation Day

On this day, September 6 (Thu), representatives from the sponsors of all the college's Clinic projects come to campus to meet with with their project teams. There will be a formal general meeting as well as smaller social events.

A Note on Confidentiality Agreements Some sponsors may want you to sign additional forms, such as nondisclosure agreements (NDAs) or patent acknowledgment forms. If you have not already discussed any additional forms with your advisor and the Clinic director, you *must* consult the Clinic director before signing them. If you sign such agreements without consultation, you may be committing the Clinic program, the department, or the college to something that we are not prepared to agree to. You may also set a bad precedent by encouraging the sponsor to make new demands without prior notice on future Clinic teams.

Even if you have discussed the forms in advance and the Clinic director has accepted the need for the additional terms, you should read them carefully and discuss any questions you might have with your sponsor's liaison, your advisor, your teammates, and the Clinic director before you sign them.

You should never feel pressured to sign a document without taking the time to read it and think about what it is asking from you. If your sponsor insists that you sign a form right then, excuse yourselves and seek out the department's Clinic director for advice. Remember that you have rights, too, and that your sponsor may demand that you surrender more of your rights than you may wish to; you always have options.

2.2 Assembling and Negotiating the Statement of Work

Before starting in on a project, your team is expected to develop a *Statement of Work*—a document that spells out what your team will do, what your sponsor will provide, and other details about your work. Your team's completed Statement of Work is submitted to your advisor and then to the Clinic director for approval. After review, the Statement of Work is shared with the sponsor's liaison, and your team, including your advisor, will negotiate any adjustments that are necessary with your sponsor's liaison(s). More details about writing such a document and how to negotiate the details are provided in Chapter 5.

There are a number of deadlines and events associated with the Statement of Work process.

The Marathon Push

The "Marathon Push" (see Section 5.2) is a self-guided crash course during which you and your teammates get up to speed on the sponsor's project proposal and get to know one another. It's a great and formative experience in the early days of the project.

During the Marathon Push, you will be producing the core of your Statement of Work document for review by your advisor. Once your advisor approves the content of your Statement of Work, it goes to the Clinic director for review and subsequently to the sponsor for approval.

Writing and Review of the Statement of Work

Chapter 5 describes how to write your team's Statement of Work. Once completed, your Statement of Work will first be reviewed by your peers, who will provide you with feedback that you should incorporate before submitting your Statement of Work to your advisor.

The deadline for submitting your Statement of Work to the Clinic director is September 18 (Tue); peer-review comments will be due back to you (and for the Statement of Work you review) on September 21 (Fri).

When both your team and your advisor are satisfied with the content of your Statement of Work, it must be submitted to the Clinic director for review and comment. When your advisor and the Clinic director are satisfied that all the necessary changes have been made in accord with all previous comments from the advisor and director, the document is considered complete and will be sent to the sponsor. **The deadline for completion of the final draft of your Statement of Work is September 28 (Fri).**

Negotiations for Your Statement of Work

Because the Statement of Work is a form of contract, both your project team and your project's sponsor must agree to its terms. Negotiations between your team and your sponsor's liaison(s) will take place during October.

The final version of your Statement of Work is submitted to your sponsor through the Clinic director and the department's Clinic Coordinator with a signed cover letter. The cover letter asks the liaison to approve the Statement of Work by signature on the letter or to return comments by a specified date; otherwise the Clinic program will assume the sponsor's tacit approval. This clause helps to ensure that your team will not be prevented from starting

work by a tardy response. Section 5.6 has more details on the negotiation process.

2.3 Sponsor Site Visits

Clinic teams typically visit their sponsor's work site twice during the course of a project: once at the beginning of the project to get acquainted and once at the end of the project to present the team's final results directly to the sponsor's managers and scientists. More visits may be made when appropriate and necessary.

The timing and details for these visits should be arranged directly by your team and your sponsor's liaison. The costs associated with making these visits (airfare, lodging, meals, etc.) should be estimated and included in your project's budget (Chapter 6).

Important Note

If your team needs to rent a car as part of a site visit, you will need to follow the rules for vehicle insurance. See Section 6.5 for details.

2.4 Presentations

Learning how to present your work to an audience is an important part of the Clinic experience. Throughout the year, there will be several formal and informal practice presentations that you will make, sometimes for just your fellow math Clinic participants, sometimes for a larger college-wide audience. Your presentation experience will culminate with your Projects Day presentations to sponsors, faculty, staff, fellow students, parents, and the general public.

Chapter 8 gives you an overview of what we expect from you when preparing and giving presentations.

Informal and formal practice presentations are scheduled for the Tuesday Clinic meeting time slot during both the fall and spring semesters. Informal presentations to the Mathematics Clinic participants take place during the fall semester, on November 20 (Tue). Formal presentations, including a dress-code requirement (see Section 8.6) take place during the spring semester.

2.5 Workshops

Workshop presentations related to Clinic project execution and management may be scheduled for Tuesday meetings when no other mathematics Clinic talk has been scheduled; see the calendar for topics and dates. Attendance is mandatory.

Past talks have included how to give and assess others' talks; how to weigh and deal with ethical considerations that may arise during a project; how to write a report; and how to manage a project.

2.6 Your Midyear Report

At the midpoint of the year (the end of the fall semester), your team will be expected to write a *midyear report* detailing your research to date and the progress that you have made towards your project's goals.

Chapter 7 includes a great deal of detail about how to write your midyear (and final) reports; what to include in your reports; how to format your report; and so forth. Appendix A discusses the workings of the hmcclinic document class; Appendix B discusses installing the class for use on your own computer; and Appendix C discusses some frequently seen LaTeX and BibTeX problems to avoid in writing your report.

Two important dates to keep in mind when working on your midyear report are

1. November 28 (Wed), when your advisor-approved draft is due to the Clinic director; and
2. December 14 (Fri), when the final draft of your midyear report will be due.

Important Note

You will need to submit both an electronic copy of the final draft of your midyear report (to your advisor and the Clinic director) and a single printed copy to DruAnn Thomas.

2.7 A Break: Lab Priority Given to the Mathematics Contest in Modeling

Harvey Mudd College generally has several teams participate in the international Mathematics Contest in Modeling (MCM), in which undergraduate

student teams use mathematical modeling techniques to tackle one of several real-world problems presented by the contest organizers and produce a formal report describing their approach and results.

The MCM takes place over an intensive long weekend in February, generally from 5:00 P.M. on a Thursday until 5:00 P.M. on the following Monday. During that time, the mathematics Clinic traditionally yields the use of the Clinic workrooms for MCM teams. See https://www.math.hmc.edu/mcm/ for more details.

If you are not participating in MCM, you are, of course, welcome to continue to work on your Clinic project using other departmental computing resources or your own resources.

This year's Mathematics Contest in Modeling is scheduled for January 24–January 28.

2.8 Wrapping It All Up: Projects Day and Beyond

Projects Day (May 7 (Tue)) is the *great day* for all Clinic teams!

You will (we hope) feel pride and accomplishment as you present the culmination of your team's hard work throughout the year to sponsors (yours and other teams'), your parents, your fellow students, staff, faculty, and the general public.

Your posters will be on display for part of the day, and you will make your formal oral presentation several times during the day to allow visitors to attend as many talks as possible.

There are several overlapping but equally significant pieces of work that you'll need to work on simultaneously. For each item, remember that you will be expected to have obtained your advisor's approval before it can be submitted to the Clinic director, and that the clinic Director must approve each item. Be sure to plan accordingly (i.e., you must have completed your work *before* the dates listed here). Chapter 9 covers the requirements in more detail.

2.8.1 Final Report

You should expect your final report to go through several rounds of drafts, review, and revision with your advisor and the Clinic director.

Dates to keep in mind are:

April 15 (Mon) Draft of final report due to advisor. (Ideally, you will have been working with your advisor all along, so this date may be only a formality.)

April 24 (Wed) Advisor-approved draft of final report due to Clinic director for review.

May 10 (Fri) Final version of your report is due to your advisor and the Clinic director. (See Section 9.4.)

May 10 (Fri) *Printed copies* of your report (Section 9.4) and CD-ROMs (Section 9.5) delivered to DruAnn Thomas. PDF copy of report also sent to DruAnn Thomas by e-mail.

2.8.2 Poster

You should be working on your poster at the same time you're working on your final report. Sections 8.4 and 8.5 have useful advice and resources for preparing your poster.

April 5 (Fri) Advisor-approved poster draft due to Clinic director for review.

April 16 (Tue) Once the Clinic director has given you feedback and you've made the changes, you'll need to get your liaison(s) to approve the final version of your poster (e.g., to ensure that there are no intellectual property issues).

Your final, liaison-approved poster must be submitted to Molly Reeves for printing by 5:00 P.M.

May 3 (Fri) Poster session for Alumni Weekend. Your poster will be displayed and you will be expected to be on-hand to answer questions about it. (Details to come.)

2.8.3 Projects Day Presentations

May 7 (Tue) is Projects Day. Your poster will be on display and you will spend time talking to interested viewers a couple of times. You will also be giving multiple presentations to audiences including liaisons (for your and other projects), faculty, staff, other students, and the general public.

Dress code for Projects Day is *formal*: Coat and tie for men, and suits or formal business dress for women—see Section 8.6 for details.

2.8.4 Archiving Work Products

May 10 (Fri) at 4:00 P.M. is when all of your "work products" must be complete and turned in, as well as returning equipment, settling accounts, and so forth. See Chapter 9 for details, but the key elements are

- Printed final reports and CD-ROMs turned in to DruAnn Thomas;
- E-Mailed PDF of Report to DruAnn Thomas;
- Any Clinic-owned materials used for your project returned to DruAnn Thomas;
- Electronic archive complete: copy of all computer files created for the project, plus the CD-ROM image stored in project directory on the server; Git repositories complete and turned over to Molly Reeves, who will also sign off on your. . .
- Closing checklist completed and delivered to DruAnn Thomas (by 4:00 P.M.).

2.9 Self and Peer Evaluations

In addition to developing your mathematical research skills, you will also grow in your project management, communication and teamwork skills. To provide you feedback on your progress and help us assign course grades, self and peer evaluations will take place at the end of each semester. You will receive a form on which you will evaluate your performance, as well as that of each of your team members. This form is due by December 14 (Fri)in the fall semester and April 26 (Fri)in the spring semester.

2.10 Redefining Success

One of the less frequently acknowledged aspects of research is that it's not always successful. You may find along the way that you need to redefine the project, change course, or modify (with your liaison's agreement and input) the deliverables of the project. You may be surprised to learn that even a negative result can be valuable to a sponsor if the analysis is well documented. Knowing that a particular approach to a problem isn't a viable path is almost as valuable as a working solution. And you will also almost

certainly have had other useful insights along the way. Document your approach and results carefully, keep an open line of communication with the liaison, and don't be discouraged!

Chapter 3

Computing Resources

The department provides administrative and computing support for Clinic projects, both through its general computing services (mail, fileservice, etc.) and through desktop computers provided for the use of Clinic teams.[1]

3.1 Computing Support

We have an extensive website devoted to our computing resources, how to find them, how to use them, and how to get help, available at https://www.math.hmc.edu/computing/.

Your primary human source for support when using the department's computers, services, and other computer-related equipment and materials (including LaTeX!) is our information technology analyst, Molly Reeves.

If you have any questions or concerns, please feel free to drop by the information technology analyst's office, SCTL 2404, or send e-mail to mreeves@hmc.edu.

3.2 Accounts

Department machines, including any Linux or Mac systems in Clinic labs, use your math-cluster login name and password for access. You will fill out the necessary forms for accounts during the initial Clinic meeting, but you can (and should) read the appropriate-use policy (also available in a

[1] During the Mathematical Contest in Modeling (see the Calendar on page i and Section 2.7), Clinic work is suspended and Clinic resources are made available for some of the MCM/ICM teams.

somewhat different form at https://www.math.hmc.edu/computing/policies/). The account-request form is also on the website.

3.3 Clinic Lab Systems

Each team is given two machines initially; depending on the needs of the project, additional machines might be added or operating systems changed.

Beginning in 2011, most Clinic work will be done in Clinic lab spaces on the third floor of the Sprague building (formerly Sprague Library). We will install desktop computers according to project needs. Additional machines with alternative operating systems may be available on request.

3.3.1 Operating Systems and Software

We generally install a standard set of software on all workstations, subject to operating system and architecture restrictions.

Other than the basic operating system and all included applications and utilities, the machines will have a recent TEX system (TEX Live), text/source-code editors, compilers, word-processing applications, and commercial software such as Maple, Mathematica, and Matlab.

This year (2015–2016), we're experimenting by providing Mac OS X and Windows machines, rather than Mac OS X and Linux systems. Many years ago, we had all three operating systems available, but that was in a lab space shared by all our Clinic teams.

Macs The Mac OS X systems are well-managed, and provide you with access to your departmental home directories as well as the most complete set of software we maintain.[2] Additional software packages can be installed through the Managed Software Center application or by request.

You log into the Macs with your departmental username and password.

Windows The Windows machines are independent of the department's servers, which means that accounts and passwords are separate from the department's authentication service.

While you get administrative access to the Windows machines, Windows machines do not mount home directories from the department's servers and *are not backed up*. Therefore, you must be careful about what you store on

[2]Most faculty use Macs, so our resources are concentrated towards supporting Mac OS X.

the Windows machines, and ensure that anything important is also stored in some other location (such as your department home directory, Github repository, or a USB memory stick). See Section 3.6 for more information.

Each machine has two accounts, usually `clinic`, an ordinary user account, and `clinicadmin`, an administrator account.

Important Note

Do not do your work using the administrator account on the Windows machines—use the `clinic` account for most work, and the `clinicadmin` account for installing software.

Linux The department's preferred Linux distribution is CentOS 7, a community-supported rebuild of Red Hat Enterprise Linux 7. CentOS is binary-compatible with RHEL, which means that any commercial software package that requires RHEL should also run under CentOS. CentOS is the distribution running on the department's servers.

If your project needs administrator-level access to a Linux system, we can either reinstall the Windows PC with Linux or provide a (possibly less powerful) PC. If CentOS won't work for your needs, we can also support another distribution (e.g., Debian, Fedora, or Ubuntu).

3.4 Shared Machines

As math-account holders, you are also welcome to use other departmental computers. You can access these machines remotely by logging in using the Secure Shell (SSH) protocol. See the computing website for more details.

The department currently has two compute servers that can be used by Clinic teams:

hex Sixteen-core (eight 2 GHz dual-core AMD Opteron 870 processors) parallel system. 32 GB of RAM.

pex 32-core (four 2.4 GHz, eight-core AMD Opteron 6136 processors) parallel system. 128 GB of RAM.

3.5 Printers

The department has a number of different kinds of printers, including black-and-white and color laser printers and a large-format plotter/printer used for printing posters for Clinic, thesis, and other research projects.

On Sprague Three, you have a black & white duplexing printer, bigfido, an HP LaserJet P3015x.

Additional printers may be made available over the course of the year.

A Note on Duplexing Our printers are all capable of duplexing—printing on both sides of the paper—and we *strongly* recommend that you always print your documents duplexed unless you specifically need them to only be printed on one side. (Duplex printing is the default on all these printers; you have to override this option to print single-sided.)

3.6 Storage (Disk) Space and Backups

We provide storage space on our servers for use by Clinic teams.

3.6.1 The "`/home/clinic`" Directory

Each team has a directory in `/home/clinic/`*`year`* (where *`year`* is the ending year for your project) for storing shared documents, code repositories, and so forth. Your final report materials (see Sections 9.5 and 9.6) will also end up in that directory, which will be archived for possible future use.

3.6.2 Home Directories

You also have a home directory at

`/home/students/hmc_`*`gradyear`*`/`*`username`*

for storing your documents, code repositories, and so forth.

These home directories are mounted via NFS (Network File System) on the department's Mac OS X and Linux, so your files can be accessed from multiple machines (e.g., from a desktop machine in your workroom and from a compute server).

Even if you choose to do most of the work for your Clinic project on your own machine or on systems provided by another department or organization, the deliverables for your project must be archived in the `/home/clinic` directory for your team. See Chapter 9 for more details.

3.6.3 Backups

Your home directories and other NFS-mounted directories on the Linux and Mac OS X systems are backed up as part of the department's regular backup cycle. Our tape-backup system runs nightly and goes through a complete set of tapes in roughly a month.

Important Note

The hard drives on Clinic machines are *not* backed up by the mathematics department—if you need to use space on the actual hard drives of the machines or on external drives connected to those machines (e.g., for extensive data sets), your team will be responsible for arranging backups of that data. *Do not store any important material directly on the hard drives of these machines*—use server shares instead.

3.7 Electronic Mail

Your account includes an e-mail address, *username*`@math.hmc.edu`. These addresses are set up to forward to your default (`@hmc.edu` HMC mail address.

3.7.1 Clinic Mailing Lists

The department maintains a set of mailing lists for communication between members of Clinic teams, project managers, advisors, and liaisons, (discussed in Section 4.5.2). The lists we create and who they go to are shown in Table 3.1).

Important Note

The mailing lists are created by scripts; if you have a new liaison or consultant join your project, they won't be added to the mailing lists until you ask the information technology analyst to add them to the lists.

Our Clinic lists are based on the math department's mail server; so you need to send mail to *list_name*`@math.hmc.edu`).

List Name	Members
General Lists	
`clinic-`*`year`*`-hmc`	All team members and advisors
`clinic-`*`year`*`-all`	All team members, advisors, liaisons
`clinic-`*`year`*	All team members (not advisors, liaisons, consultants)
`clinic-advisors-`*`year`*	All faculty advisors
`clinic-pms-`*`year`*	All project managers
Team-Specific Lists	
`clinic-`*`team`*	*`team`* team members
`clinic-`*`team`*`-all`	Above plus liaisons and advisors
`clinic-`*`team`*`-hmc`	Team members and advisor

Table 3.1 Mailing lists created for Clinic projects. *`team`* is your team's short name; *`year`* is the last year of your project's school year (i.e., a Clinic project starting in the fall semester of 2010 and ending in spring of 2011, will have a *`year`* of 2011.

3.7.2 Other Mailing Lists

There are a few additional lists you should be aware of.

Class Lists

Each semester, CIS sets up mailing lists for the class based on the registrar's list of students registered for mathematics Clinic (Math 193). We sometimes use these lists to send important messages to all students enrolled in Clinic, so if you join the class later on, make sure that you are added to the appropriate list.

Resource Lists

Some other lists that you might be interested in include

`linux-l@g.hmc.edu` The `linux-l` list is for discussing the use of Linux at Mudd. It's also a good place to ask for help from a collection of professors, systems staff, and fellow students.

`latex-l@g.hmc.edu` The `latex-l` list is a place to ask (and answer) questions about using LaTeX. As with the Linux list, there's a diverse

group of people who can help you out.

We have instructions for subscribing to these lists available on the department's website, at https://www.math.hmc.edu/computing/support/email/mailing-lists/.

3.8 Github Repositories

This year we are experimenting with providing Github repositories for use by Clinic teams. There is much more information about using Git in Section 4.8, including suggested clients, ways of using it, and so on.

To get started, you'll need to sign up for a Github account (at https://www.github.com/) and then fill out the form (linked to from the Participants webpage, https://www.math.hmc.edu/clinic/participants/).

3.9 Other Off-Campus Computing Resources

Because of security and privacy restrictions, computer services outside the Claremont Colleges may not be used for the project without prior permission (written or e-mailed) from the sponsor. This specifically includes "cloud" services such as Google Docs, Amazon Web Services, Amazon EC2, Dropbox, ShareLatex, Flickr, and so on.

Two important exemptions to this rule are Github, because we're hosting repositories there using Clinic accounts, and Gmail and other Google Apps, because the college has outsourced student e-mail to Google and the other Google Apps services are included under the same contract, which guarantees confidentiality and privacy.

Important Note

If your project falls under ITAR (International Traffic in Arms Regulations) or other legal restrictions there may be additional limitations on what services you can use, as well as requirements for internally hosted services and equipment.

If your sponsor is a national laboratory, defense contractor, or in the aerospace industry, you will need to make sure that any service you want to use is acceptable to your sponsor, and coordinate with the Clinic director and the department's system administrator as early as possible once your project starts.

3.10 Additional Equipment, Software, or Other Resources

Hardware and software purchases must be included in your budget, which is due on September 24 (Mon) (the same due date as your Statement of Work), and approved by the department's information technology analyst and the Clinic director if the cost is $500 or more. Thus you must begin to consider your needs and work with the Clinic staff to address those needs as soon as possible.

Your first step is evaluate your project's requirements. Consult with your advisor and the department's information technology analyst to look for ways to address those requirements—we may be able to install free/open-source software,[3] already have licenses for commercial software, or have equipment in storage that will meet your needs.

If we don't have suitable equipment or software, then the next step is to look for solutions that address your immediate needs as well as the department's preference for solutions that can be used for other purposes after the project year.

Section 6.4 has more details on the approval and budgeting process.

3.10.1 Use of Purchased Hardware or Software

We expect that any software, hardware, or data purchased by the college as part of a Clinic project will be used by other Clinic projects or general account holders after the project is completed, although access to these resources will be limited to use by the team whose budget purchased them during that year.

Also, if more than one team needs the same or similar resources, we may consider consolidating the purchases (e.g., splitting the cost of a piece of hardware that can serve more than one team between those teams; sharing a single (e.g., node-locked) software license or buying a multiseat software license) to minimize costs and maximize the potential for future use.

Commercial software or data purchased with Clinic funds will be *licensed to the department or the college*, and *may not be installed by or used by the sponsor*

[3] When possible, we prefer to have sources for RPMs or other appropriate packages pointed out to us to make installing the software easier.

unless the license terms allow such use. The sponsor may not obtain an expensive piece of software for commercial use at academic pricing through the Clinic program.

3.10.1.1 Sponsor-Supplied or Hosted Software

Should the sponsor wish to have the software for their own use without purchasing a separate license or licenses, they may be able to purchase the software license(s) themselves and allow your Clinic team to use it under the licensing terms for the project, subject to review by Clinic staff and legal counsel.

In such a case, we would work to ensure that the license terms are followed to the letter, which might mean restricting the set of machines or users that can access the software.

Another alternative is for the sponsor to obtain and install the software on their own systems, and give the team access to those systems for the purpose of running the software.

Important Note

You must discuss any software-sharing plans with Clinic staff to ensure that the arrangement is acceptable.

3.10.1.2 Sponsor-Supplied or Hosted Hardware

Similarly, should a sponsor require a piece of specialized or particularly expensive hardware, especially something that might not be generally reusable by the college in the future (an example from a past project is a high-end colorimeter), we would prefer that the sponsor either supply the hardware subject to return after the project is completed, or host the hardware themselves and make it accessible to the Clinic team.

Important Note

As with software, you need to discuss the possibility of using sponsor-supplied hardware with Clinic staff in advance.

Chapter 4

Doing the Work

Your Clinic project is a challenging six-unit, two-semester course and requires the same nine to ten hours of dedicated work per week expected for any demanding subject. It is extremely important for your team to agree to a work schedule during the first days of your project and to stick to your schedule throughout the year.

Your project's schedule will incorporate *six to seven hours of work sessions* each week for you to make progress on your project's goals as well as *three hours of meetings to share information* with other members of your team, your faculty advisor(s), your sponsor's liaison(s), and your fellow Clinic students.

Although you will spend time assembling your Statement of Work (Section 2.2 and Chapter 5), writing your midyear and final reports (Chapter 7), and preparing and giving presentations (Chapter 8), most of your time will be spent actually "doing the work"—pursuing one or more lines of research towards your project's goal. This chapter discusses some of the routine activities that make up that process.

4.1 Meetings

Typically, you'll have three hours a week set aside for team and class meetings.

4.1.1 Class Meeting

The scheduled meeting time for Clinic in both the fall and the spring semesters is Tuesday from 11:00 A.M. to 12:00 P.M. In the fall semester, this meeting time will be used for mathematics Clinic–specific activities such as project management workshops, informal presentations and critique (see

Chapter 8) and project manager meetings. During the spring semester, you will attend (or give) formal college-wide presentations during this time block.

If there are no presentations and no other event has been scheduled during the time block, your team should consider this block as an additional work session.

4.1.2 Team Meetings

Your team will typically meet as a group for two one-hour–long meetings each week; at least one of those meetings should include your faculty advisor and liaison. Your team, advisor, and liaison can work out the most convenient times for these regular meetings.

During your meetings you should share new information, review your progress, assign tasks, and so forth. Your project manager should prepare an agenda in advance of the meeting, and the meeting should follow the agenda. The meeting need not run for the full hour if everything on the agenda is addressed in less time, but you might choose to take advantage of any "extra" time to work on other aspects of your project.

For teleconferences with your liaison, your project manager should send the agenda to your liaison before the meeting, and each member of the team should take ownership of at least one topic to discuss with the liaison.

4.1.3 Project Manager Meetings

In addition to team meetings and work sessions, project managers will meet regularly with the Clinic Director to discuss management issues and to keep the director apprised of their team's progress.

These meetings may be either one-to-one or group meetings to allow the discussion of common issues. Project managers will learn that problems should be expected, and that advice and assistance is available to help solve those problems.

4.2 Work Sessions

In addition to your team and class meetings, you will need to set aside regular time slots for a total of six or seven hours per week for coordinated joint work among team members as well as individual work. These meetings should take place in your primary workspace, which may be the mathematics

Clinic lab spaces or some other space, depending on your project and the arrangements you make with the department and the Clinic director.

We suggest that you consider having two work sessions each week—a three-hour session and a second three-to-four hour session depending on your project's needs—but the exact number and length of sessions is up to you. The key is that your team should commit to disciplined and regular work sessions to ensure steady progress towards your project's goals.

Keep in mind that you will probably be sharing some resources (rooms, computers, etc.) with other teams, and you may need to be flexible to accommodate everyone's schedule.

4.3 Sharing Your Schedules

Once you have worked out your regular meeting schedule, make copies for all the team members and your advisor and post a copy in your primary workspace for easy reference.

We encourage you to use on-line calendaring systems (e.g., Zimbra, Google Calendar, Yahoo! Calendar) to keep everyone organized.

4.4 Medical Conditions

If you have a medical condition such as hypoglycemia, hyperglycemia, diabetes, epilepsy, pregnancy, or any other medical condition which may cause sudden loss of consciousness, please inform your instructor of the condition at the beginning of the semester or as soon as you become aware of the existence of the medical condition.

You may also want to share that information with your teammates, so they will be aware of a possible problem and know what to do to help you.

4.5 Recording and Communicating Your Work

In addition to your regular meetings, you should expect to record your progress on the project in a *work journal* or *research notebook* as well as share information with other members of your team, your advisor, your liaison, and the Clinic Director. There are a number of options available, some of which are described here. The final set of tools that your team chooses may be negotiated within your team and with your advisor, liaison(s), and the Clinic Director.

4.5.1 Work Journal or Research Notebook

Each team member should maintain a *work journal* or *research notebook* to record their work on the project.

Journals or notebooks will help you keep track of your own progress; may be used in determining grading (see Section 10.1.3); and may be necessary should your sponsor wish to obtain patents for any innovations that may have been discovered during the course of the project. Appendix E discusses the details of how you should use your notebook; Purrington (2011) is a very good overview, with specific advice on inks, paper, and so forth.

4.5.2 Electronic Mailing Lists

Section 3.7.1 provides details about the electronic mailing lists that will be set up for your team. Table 3.1 lists the mailing lists for your team. Use these lists rather than your own personal lists.

Important Note

It is your responsibility to check and respond to e-mail messages on a timely basis; most communication from Clinic staff will be conveyed via e-mail.

4.5.3 Liaison Teleconferences

One of your weekly meetings will include a teleconference with your team's liaison. During these calls you will keep your liaison apprised of your progress and seek their advice on problems that you have encountered. Appendix H discusses teleconference etiquette.

4.5.4 Personal Telephones

Page v lists telephone numbers and other contact information for Clinic and departmental staff. You should also make sure that everyone on your team has access to contact information for your fellow team members, liaison(s), and other important people and places. Your notebook (Section 4.5.1) are good places to record this information.

4.5.5 Other Communications Tools

There are many other communications tools available today, including instant messaging (from various providers), Internet Relay Chat (IRC), and more. Feel free to use whatever tools work best for you and your team, but make sure that what you discuss is appropriate for the particular medium—be aware of the confidentiality limits to what you can discuss in public and keep your discussions well within those limits (see Section 2.1.

4.6 Working Together

The following techniques can help you and your team to make the most effective use of your scheduled time:

- Assign individual team members to learn about a specific topic, and make a short oral report to the team in a meeting. A written version can go into a draft of the midyear or final report.
- Break into pairs to do specific tasks.
- Seek opportunities for transferring skills between team members (e.g., coding, HTML, LaTeX, MATLAB, SAS, or other software packages).
- Revisit your working assignments from time to time and consider whether reassigning some work, switching groups, or other changes might improve your progress.
- Recruit appropriate consultants/experts. Good candidates include Mudd or other Claremont Colleges faculty members from other departments, or a student from an earlier relevant Clinic project.
- Work steadily and conscientiously on schedule; keep in touch with your teammates; and keep good notes on all the work you do, whether successful or not.
- Remember that a negative result that is well-explained and understood is still a *good research result*.

4.7 Library Resources

In addition to the Claremont Colleges–wide library resources available through Honnold-Mudd Library, the department has an extensive collection

of mathematics books stored on the third floor of Sprague (i.e., the same location as you Clinic workspaces).

These books are cataloged in the LibraryThing web catalog, available at http://www.librarything.com/catalog/hmcmath. While most of these books do not circulate, those tagged as "clinic" (which can be viewed at http://www.librarything.com/catalog/hmcmath&tag=clinic) can be checked out for use by your team through DruAnn Thomas.

4.8 Using Version Control

Each team will be expected to maintain the LaTeX code for their reports and any other code they write in a *Git repository*. Git is a *version-control system*, or VCS.

In principle, you could manage files in a shared group project (e.g., in a Google Drive or Dropbox directory) without ever using a VCS, but version-control systems aim to solve two problems that arise with other methods of sharing code and document files:

1. If you modify the contents of a file (e.g., source code or documents) the old version disappears as soon as you hit "save". Sometimes you might want to either undo changes or check to see what has changed over time, and unless you had enough foresight to make backup copies at the right time(s), restoration and comparison can be difficult or impossible.

2. If more than one person is working on a project and directly sharing the same set of files, it can be difficult to figure out who changed what and when. Worse, if two people try to edit the same file at the same time, whoever hits "save" last will overwrite the file and any changes made by the others will be lost.

   But if everyone has their own separate copy of the files, trying to assemble a consistent copy of the code with all the latest and best can be complicated and time consuming.

VCSs solve the first problem by maintaining a *repository* of files. When a file is saved ("checked in" or "committed") to the repository, not only is the new version of the file stored, but all previous versions of the file are remembered in case you want to revisit them.[1] Thus, you can always ask

[1]To save space, the repository generally stores only the *differences* between each successive

the repository for a previous version of a file, or for the differences between any two versions of a file.

Traditionally, VCSs have had a centralized, shared repository, where all changes by any user are stored. Users could get a copy of the repository—a *working copy* (see Section 4.8.5)—make their changes in their own copy, and then put those changes back into the shared repository.

The shared repository meant that users would have to be careful to avoid overwriting parts of files changed by other users—often, two people working on different projects would need to touch some of the same files. When they were done, and copied their changes back to the repository, the person who checked in their changes last would either have their changes replace changes made by the other person, or result in conflicts, where two people had modified the same bit of code. Those conflicts would have to be resolved, which means going through the code and ensuring that the result still does what both people intended it to do.

In the past we've used Subversion, a fairly new (ca. 2000) centralized VCS meant to balance the traditional way of doing things while still addressing many of the limitations of those systems.

Unlike Subversion, Git is a *distributed version-control system*, which can be used without a central repository.

Git allows you to have multiple independent working copies (one or more for each person), and each of those working copies is a complete repository, can have new working copies checked out of it ("cloned" in Git-speak), and can be used to restore a centralized repository if something happens to it.

Even better, because Git repositories are independent from one another, you can do your work when you can't get access to the central repository, and still be able to do all the commits you want to, pushing those changes up to the shared repository later on. That independence also means that you can work on a feature on your own until it's done before you commit it to the shared repository.

(Chapter 1 of *Pro Git* (Chacon and Straub, 2015: 27–31) gives you a nice overview of the history of VCSs.)

4.8.1 Git Resources

Some people see Git as much more complicated than more traditional VCSs such as Subversion. In some ways that's true, in that Git adds some

version of the file; by combining all this information in the right order, any particular version of the file can be reconstructed.

additional layers compared to a centralized VCS, but in return Git gives you a great deal of flexibility that you don't get with a centralized repository, probably the least of which is being able to work without access to the centralized repository (or to be able to complete a whole chunk of work, with commits, without having to push it to the shared repository until you're done).

Luckily, Git has a lot of good documentation, and, because it's so popular—it was originally written by Linus Torvalds to manage the Linux kernel, and thus immediately had a vibrant community (Chacon and Straub, 2015: 31)—there's lots of information out on the Web. Chances are that someone has run into the same issue you have, or wanted to do the same sort of thing you want to do before, and the answer is a Google search away.

Here are some useful links:

- https://git-scm.com/: Git's home, with lots of links to documentation, tools, and more.
- https://git-scm.com/doc/: Git's official documentation, including Chacon and Straub's *Pro Git* book (2015), which is available under a Creative Commons license and can be purchased as a physical book, or downloaded for free in various e-book formats (PDF, ePub, mobi, HTML) from https://progit.org/.[2]

  Because *Pro Git* is free (and both an excellent introduction and reference), you should just download a copy for yourself to keep and use (contrary to the general advice not to buy/obtain e-books over physical books given in Chapter 6).
- https://stackoverflow.com/questions/tagged/git: Latest questions about Git on https://stackoverflow.com/, a technical support site for developers.

4.8.2 VCS Tools

Command-line tools are included with most or all Linux distributions and with Mac OS X (with the (free) XCode developer tools installed). For Windows, you can install Git from https://git-scm.com/download/win/.

If you're not comfortable with the command line, or you want to get a visual overview of your code, there are a number of different GUI clients for Git. You can probably find more by searching online, but here are several well-known and well-regarded options:

[2]Further references to *Pro Git* will use page numbers taken from the PDF.

gitk Included with Git (free; cross-platform)

Github Desktop Github Git client; specifically optimized for working with the Github service (free; Mac/Windows; https://desktop.github.com/)

SourceTree Git/Mercurial client from Atlassian (free; Mac/Windows; https://www.sourcetreeapp.com/)

TortoiseGit Integrates into Windows File Explorer (free; Windows-only; https://tortoisegit.org/)

GitX Another GUI (free; Mac-only; http://gitx.frim.nl/)

4.8.3 Provided Repositories

Your team will initially have two repositories, code and reports. Depending on how your project goes, you may need or want additional repositories,[3] and you can have a conversation with your advisor and the department's system administrator about setting those up.

We recommend that you create a directory called `sow` in your reports repository, and work on your Statement of Work there. When you're done with your Statement of Work (or ready to start working towards your midyear report), you can use `git copy sow midyear` to get a copy of your Statement of Work files ready to modify for your midyear report.[4] You can do the same sort of thing for your final report.

If you need to write computer code, make directories inside your code repository, and work in those subprojects unless you really need to have separate repositories.

If your team does decide it needs additional repositories, consult your advisor and the department's information technology analyst. We might have another way to do what you want, grant you additional repositories, or even end up deciding to add similar repositories for all of the year's projects.

Important Note

[3]Unlike Subversion, Git doesn't allow you to check out subtrees, which makes it more difficult to have more than one project in a single repository. There are still good reasons to have a single repository for subprojects—such as being able to easily share code between those subprojects—but Git is heavily oriented towards the "separate repository for each project" approach, and sometimes that is the best way to do things.

[4]You could also create new branches, but that approach is harder for some people to follow.

If you fork your Clinic repos within Github (or a similar service), make sure that they are *private* repos. Your sponsor expects confidentiality, and so do we. (Even if your sponsor thinks your repository should be publicly available, you should still talk to your advisor and the Clinic director first.)

4.8.4 What Goes in the Repository?

Not everything in your working copy belongs in your repository. Many programming language interpreters and compilers generate files in the process of converting code to an executable application—object (`.o`) files for C and C++; `*.pyc` files for Python code, and so on.

LaTeX is a particularly messy application, as it generates a large set of "auxiliary" files to allow it to pass information to successive runs as well as log files to help you understand what it's doing and debug problems. You should not include any of these files in your repository.

Important Note

A good rule of thumb is that if you didn't create a file by hand (e.g., it was generated by running a program or you downloaded it), you shouldn't include it in your repository.

4.8.4.1 Things to Check In

- LaTeX source code (`*.tex`) files
- BibTeX bibliography (`*.bib`) files
- *Final* images or diagrams (`*.pdf`, `*.jpg`, `*.png`)
- Computer program source code

4.8.4.2 Things to Exclude

- Typeset files (e.g., `*.pdf` files from `*.tex` files you created and maintain)
- LaTeX auxiliary and log files (e.g., `*.aux`, `*.bbl`, `*.blg`, `*.brf`, `*.bux`, `*.dvi`, `*.fff`, `*.idx`, `*.ilg`, `*.ind`, `*.lof`, `*.log`, `*.lot`, `*.`

`out`,
`*.pdfsync`, `*.ps`, `*.rel`, `*.toc`, `*.ttt`)

- Compiled programs and intermediate files (e.g., library files, object files, bytecode files)
- Data files (see Section 9.5.3)

Being careful to not accidentally add and commit one of these files is a pain; luckily, Git provides a mechanism for ignoring these files completely—the `.gitignore` file.[5] (See next section.)

4.8.4.3 `.gitignore` and `exclude` Files

Git provides several ways of specifying files to ignore

1. A global, per-user, `.gitignore` file;
2. A shared, per-project `.gitignore` at the top level of a repository (that is checked in and out like any other file in your repo); and
3. A per-user, per-repo ignore file in *`repository root`*`/.git/info/exclude`,

all using the same syntax (shell "globs" or matching patterns; see `man` *`your_shell`*.[6] for details).

You'll find a sample `.gitignore` file on the Clinic Participants page (https://www.math.hmc.edu/clinic/participants/). If you save this file to your home directory (notice the `.` at the start of the filename!), it will be used by Git for any and all repositories you access on machines that use that home directory.

See Chacon and Straub (2015: 50) for more details.

[5] You can still make Git do things with the files, but it takes extra steps, and they don't show up every time you do a `git status`.

[6] You can find out what shell you're using by running `echo $SHELL` and then looking at the last part of the path (e.g., if `echo $SHELL` gives you `/bin/tcsh`, your shell is `tcsh`, and you can get details on pattern matching by running `man tcsh` and searching for "pattern" or "glob").

4.8.5 The Working-Copy Model

To work with any of the files in the repository, you must check out ("clone") your own *working copy* of the files in the repository. You can then modify any of the files in your working copy as you see fit. While you work in your working copy, nobody else can see the changes you are making. If you only want to look at the files, you can simply delete your working-copy directory when you're done and no one will notice. But if you make changes and want other people to see them, you will need to *commit* your changes and then *push* those changes back to the shared repository. Similarly, you can update your working copy to incorporate any changes that your teammates may have committed to the repository.

Pages 34–35 and Chapter 2 of Chacon and Straub (2015) introduce Git's use of working directories.

4.8.6 Git Workflow

The main adjustment you will have to make when you work with a VCS is adopting a *workflow*—a sequence of actions that you take to make sure you have the latest copy of everyone else's work when you start and that you've shared your work once you've finished.

Pages 34–35 and Chapter 2 of Chacon and Straub (2015) explain Git's workflow in detail, but the following list gives a quick overview.

Getting a Repository

> **`git clone` *`repo URL`*** Getting an initial working copy of the repository on your machine, or
>
> **`git init`** Create a new repository in an empty directory or from a set of existing files.
>
> (Chacon and Straub, 2015: 43-45)

Working Doing your work, which could include

> **Editing** Changing existing files
>
> **Adding** Adding new files
>
> **Deleting** Removing files (if in the repository, using `git rm` *`files`*)
>
> **Renaming or Moving** Renaming or moving files from one place to another (if in the repository, using `git mv` *`files`*)

Commmiting Once you have a task completed, or you've gotten to a good stopping point (in most cases, your code should still build and LaTeX documents should typeset), you can *commit* (or *check in*) your work by

Staging Adding the changes to the stage prior to committing with `git add` *`changed file(s)`*

Committing Updating your working copy with the changes you've made, including a meaningful commit message (see Section 4.8.7) with `git commit -m` *`message`*

(Chacon and Straub, 2015: 45-58)

Sharing Your Work

`git pull` *`remote branch`* Updating your *working directory* from the repository so you have the latest set of checked-in files. (The "remote" is a name for a remote shared repository. See Section 4.8.6.1 for an explanation of the "branch" argument.)

`git push` *`remote branch`* Sending your changes to the shared repository.

(Optionally, you can use `git fetch`/`git rebase` to get changes from upsteam. See Chacon and Straub (2015: 81–124) for more.)

In general, as you work, you may want to update your working copy at convenient moments to make sure that your working copy stays current with any changes that other people have made. Similarly, if you're working together on a particular project (e.g., writing a chapter of your final report), you will probably want to commit your changes on a regular basis so that others will be able to see and work with them. At the very least, your last action at the end of a work session should be committing your changes, at least to your local repository. (You can, for example, do a "work in progress" commit, and revert it later.)

When it comes to pushing changes to the shared repository, keep in mind that it is antisocial to commit changes that will break things for other people (e.g., source code that doesn't compile; LaTeX code that doesn't typeset properly).

4.8.6.1 Branching and Merging

One of the features that makes Git popular is "cheap branching"—if you want to work on a particular thing without getting it mixed up with other

work or upstream's code, you can create a *branch*, do your work there, and, once it's done, merge that work into another branch (often `master`, the default branch in every new Git repository).

Other VCSs have branches, too, but Git allows you to have both shared branches (branches that appear in the shared repository and can be downloaded and contributed to by others) and local branches that only appear in your repository. (Subversion, for example, only has shared branches.)

Combining branches is called *merging*, and pulls the changes from one branch into another, making the changes that don't interfere (*fast-forwarding* (2015: 92–93)) and marking *conflicts* for you to address. (2015: 96–98)

Chapter 3 of *Pro Git* covers branching and merging in much more detail.

4.8.7 Writing Good Commit Messages

When you commit a set of changes to the repository, you should include a message explaining what you've done. You don't need to explain every change in detail, as the exact changes can be viewed with the `svn diff` command, but you should include enough information that someone looking at the log file will have a good sense of what was changed and *why*.

You can often get away with a single short message, such as

```
Made changes specified by our advisor.
```

but sometimes the changes you've made may require more explanation, especially if you changed a lot of files or made some complicated changes that aren't obvious at a glance.

In such cases your commit message should generally have

1. A one-line summary
2. An empty (blank) line
3. Additional sentences or short paragraphs explaining the changes in more detail

Git's `log` subcommand only prints the first line of your commit message, so making it meaningful is especially important!

Important Note

Be Professional! When you're writing your commit messages, remember that you and your team members are not the only people who will see them. Your advisor, the Clinic director, your liaison, and other people from the sponsor's organization or future Clinic teams may need to look at your code, and may review your log messages. So keep your log messages clean—it might feel good to swear after struggling with a stubborn problem, but remember that anyone looking at your repository will see what you wrote, and know you wrote it.

Chapter 5

Preparing Your Project's Statement of Work

Your Statement of Work is *the* key document for your project, stating your understanding of your research project at the start of the year, and it is vital that you prepare it well. Your team's initial draft will be reviewed by both your faculty advisor and the Clinic director.

That revised version of your Statement of Work will then be shared with your sponsor. Your team and your liaison(s) will discuss the terms of your Statement of Work and possibly modify it as a result of these negotiations. Once agreed to by both your team and your sponsor's liaison, your Statement of Work will serve as the foundation for your project.

Keep in mind that research is unpredictable, so be sure to allow for various contingencies and be reasonable in your negotiations with the sponsor. This chapter describes the content and structure of a Statement of Work and offers suggestions for preparing your Statement of Work and discussing it with your liaison.

5.1 What Is a Statement of Work?

Your Statement of Work is your team's commitment to your sponsor. The purpose of the Statement of Work is to outline the team's understanding of the problem, possible solution approaches and the final set of deliverables. It also includes a statement of what direct support—if any—you will receive from your sponsor, including items such as software, hardware, data, or written materials. It will also serve as a commitment by your sponsor to not

demand more from your team than you agreed to do.

Your Statement of Work amounts to a contract between your team and your sponsor, so it is important for you to prepare it carefully and to conduct your negotiations with your sponsor's representatives with caution. Never use technical terms that you do not throughly understand in your Statement of Work; otherwise you may discover too late that you have promised to do the impossible. Make sure that you understand your Statement of Work very well before you open discussions with your sponsor's liaison.

A project's Statement of Work is typically around ten pages long when typeset using the hmcclinic LaTeX class file (see Section 5.5). The document spells out what your team is agreeing to do and, in turn, what you need from your sponsor and when you need to receive it to get the work done.

Important Note

Do not give your liaison a copy of your draft Statement of Work until *after* the Clinic director has had a chance to review it and discuss it with your project manager and faculty advisor. Be sure to keep the director in the loop to help you avoid pitfalls.

Not giving your liaison copies of your working document doesn't mean that you shouldn't discuss your ideas with them before submitting the completed document. As you will need your liaison's final approval, you should do your best to find out what your sponsor is expecting from your team.

5.2 The Marathon Push

On Sponsor Orientation Day (September 6 (Thu)), you and your teammates will meet for the first time with your faculty advisor and your sponsor's liaison(s) to discuss the problem and get to know one another socially.

For the next three weeks, until the Statement of Work is due on September 18 (Tue), you will have the chance to get your teeth into your project during the *Marathon Push*.

During this period, your team will work together to tackle the goals outlined in this chapter. We expect that you will be working on your own, although you are free to consult your faculty advisor at any mutually agreed times.

Your advisor-approved draft Statement of Work report will be due for peer review on September 18 (Tue). Once you have incorporated revisions suggested by your peers and your advisor, the advisor-approved draft Statement of Work is due to the Clinic director on September 28 (Fri).

During the marathon push, we expect the team to spend an extra fifteen hours, on top of regularly scheduled work sessions, to complete the background research necessary to write a strong Statement of Work.

What's the Purpose?

Everyone on your team must get up to speed on the project as quickly as possible, so that you can begin making good progress on your project as early as possible. This marathon push is intended to help you jump right into the immediate tasks that must be accomplished; specifically,

- A literature search (books, papers, websites, past Clinic reports, etc.)
- Formulating possible methods for achieving a solution to your problem
- Discovery of the mathematical requirements of the problem
- Beginning to think about and understand the potential impacts of your project from social, economic, or political points of view

The marathon will also help you prepare to write your Statement of Work and to put together your first oral presentation.

Why Such a Short Time?

You will already have several days and at least one discussion about your problem under your belt before you begin the marathon, so you should have some ideas about how you want to proceed before the marathon push starts. And, of course, we recognize that you will have many other demands on your time, even this early in the semester. But this short period of intensive focus can help your team bond and give your project a real kick-start.

5.3 Who Will Read Your Statement of Work?

When writing any document, you should consider the audience for the document. Your expected audience will largely determine the style and vocabulary of your final document.

Your Statement of Work is not meant solely for experts. It will be of interest to several parties of differing backgrounds and knowledge, not just your liaison, your advisor, and yourselves. For example, your sponsor's management, who funded the project, will need to understand what they paid for and be able to justify the expenditure to *their* managers. If your work is successful and generates interest in implementing the results or in continuing the line of research, managers will once again be tapped for funds, and a later project team, who are not necessarily experts at the outset, will need to come up to speed on the project just as you did. A clearly written Statement of Work, along with your final report, can be used by them as their starting points.

At the same time, you should recognize that not everything can be explained for the lay reader. As mathematicians, we know that mathematics is itself a specialized language, and where formulas are required there is no getting around them. Your choice of language is a matter of judgment. Just keep in mind that your Statement of Work must serve readers of multiple backgrounds and purposes. Define all acronyms, explain the technical terms, avoid esoteric language and use plain English where possible.

5.4 Content and Layout of Your Statement of Work

Your Statement of Work should contain a description of the project as you and your team imagine it will progress. On the broadest level, you will probably want to include sections that roughly correspond to the following elements:[1]

Problem Statement A succinct restatement of the problem as presented to you by the sponsor.

Background Some information about the sponsor, what they do, why they're interested in the problem, and so forth.

Goal The overall, long-range, end result that your research is *aimed at achieving*; where you are ultimately trying to get. Stating a goal does not mean you believe you will achieve it during this project; it is the grand view towards which you strive. For example, the goal of HIV research is to find a cure for AIDS.

[1]See Section 5.4.2 for more detail about some of these elements.

Objectives The specific things that you will *try to achieve* during the course of your project; the immediate targets of your research. Your objectives spell out how you have parsed the problem of the ultimate goal into the smaller pieces that you will work on.

The objectives set the practical limits on your work. They point to what your team can reasonably expect to achieve. The objectives should clearly fit into and work towards the long-range goal.

Literature Search A description of scholarly work that is related to your problem.

Possible Approaches At least two ideas for possible solution methods, with outlines of the potential strengths and weaknesses of each method.

Tasks The specific things that *you will do* in order to achieve your objectives. The tasks drive your determination of what skills and other resources (e.g., data, software, hardware, written materials, work environment) will be needed for your project. If any of these resources must be supplied by the sponsor, you will need to specify these items in your Statement of Work (see Section 5.6).

Schedule A list of dates and times that specifies *when you will finish major parts of your research* and *provides a timetable for completion of deliverables*. Internally, you should maintain as fine-grained a schedule as you need to keep your team coordinated and on track, but in your Statement of Work it is best to make the schedule and list of deliverables as modest as the sponsor will allow.

Milestones A *list of specific accomplishments* that you may use *to mark progress and maintain pace and coordination* within your project.

They are used to help your team to stay on track and to determine the success of a chosen line of attack on your problem. A list of milestones may or may not be included in your Statement of Work, but you should definitely think them through for your own use as you plan your project and Statement of Work. They are checkpoints for you (and for your sponsor, if they are included in the Statement of Work), *not deliverables*.

You may want to specify major milestones in your Statements of Work to indicate what you would do if your research leads to the conclusion that some objective cannot be accomplished. For example, "If by

some date we have found it impossible to achieve X, then we will begin Y." Research is exploration of the unknown, so you may bump into an intractable obstacle and need to work around it—you can't know everything ahead of time. Give some thought to these concerns and try to allow for milestones that allow you to judge how well you are proceeding towards your objectives and deliverables and what you need to do to proceed effectively in the event you don't meet a milestone.

Deliverables The things *you promise to deliver* to the sponsor. For a mathematics Clinic project, these items include a midyear and final report, a poster, and a final presentation on Projects Day. They may also include site visits to the sponsor (usually one near the beginning of the project to get acquainted with the sponsor, and one after Projects Day to present your work at your sponsor's location); software; hardware (in some cases); written results of literature searches; *white papers* (i.e., written background information on such things as plans, methods, or concepts prepared for internal use); and so forth. These additional items are to be decided by you in consultation with your liaison.

Your Statement of Work need not use the vocabulary terms introduced above, but it should address the issues that they cover. And your Statement of Work need not be as fine-grained as this example implies. For example, your team might set internal milestones as a way of maintaining pace and coordination but not include some or all of them in your Statement of Work. Your Statement of Work is like a recipe—getting the right ingredients and the right amount of each ingredient is a bit of an art.

5.4.1 Sample Statement of Work Structures

Figure 5.1 shows four sample templates for statements of work taken from recent HMC Clinic projects.

Note that none of the projects represented by these four templates called for the sponsor to provide any software or other data, materials, or services required to complete the project. For this reason no contingency planning was required in case of failure by the sponsor to deliver promised material.

If your project *does* require your sponsor to provide some important element, make sure that your Statement of Work includes a last acceptable date for delivery of the material and contingency plans that will allow you to proceed in the event of a failed delivery. Note that this category could

Template 1

- Abstract
- Background
- Problem description
- Approach ("time permitting" clause for some work)
- Schedule
- Milestones
- Deliverables ("time permitting" clause for some work)

Template 2

- Introduction
- Problem statement
- Mathematical background
- Computing background
- Possible solutions and project objectives
- Deliverables
- Timeline

Template 3

- Project background
- Goals
- Proposed mathematical approach (options specified in case of failure)
- Objectives
- Optional objectives ("time permitting")
- Deliverables
- Milestones
- Work flowchart
- Schedule

Template 4

- Abstract
- Problem background
- Problem description
- Approach (Team reserved right to choose software compatible with sponsor)
- Deliverables
- Timetable
- Team members

Figure 5.1 Templates for Statements of Work.

include, for example, consultation provided by the sponsor on the use of special equipment or software.

5.4.2 Getting More Detailed

Throughout your Statement of Work, you should

- Define your terminology—avoid undefined jargon or acronyms.
- When there are questions that need answers to allow the project to proceed, give the dates by which they must be resolved.
- Include modest and reasonable objectives.

The following sections include a few more suggestions for the content of specific sections in your Statement of Work.

5.4.2.1 About the Sponsor

In your introductory section (e.g., "Abstract," "Introduction," "Background"), you might want to include one or more of the following:

- A brief description of the sponsoring organization
- Major product line(s) or what they do
- A brief (abstract) description of the project

But don't go overboard. Your concern should be explaining enough about the sponsor to be sure that *you* (and maybe a naïve reader) understand why they're interested in the project, not parroting your sponsor's marketing material.

5.4.2.2 Deliverables

Your list of deliverables should include details such as

- Site visits (to be arranged)
- Oral presentation(s)
- Midyear report
- Final report
- Projects Day presentation and poster
- Software (if appropriate)
- Documentation for software or hardware (if appropriate)
- Specify sponsor-approved operating system or computing platform

5.4.2.3 Contact Information

It may be useful to include a table of participants and contact information (e.g., physical and electronic mail addresses; telephone numbers; instant messaging nicknames; URLs for wikis or other shared document systems or websites).

Important people to include are

- Team members (be sure to specify the project manager)
- Faculty advisor(s)
- Sponsor's liaison(s)
- Any consultants
- The Clinic director and relevant departmental staff

5.5 Formatting Your Statement of Work

LaTeX is required for the Statement of Work and all subsequent Clinic reports.

To help you format your Statement of Work, the department provides a LaTeX document class, `hmcclinic.cls`, that can be used for your Statement of Work as well as for your midyear and final reports.

The document-class file takes care of most of the basics of formatting your Statement of Work, including formatting a title page and setting margins, fonts, and other common design elements of a document.

Using LaTeX gives you powerful text and mathematics typesetting and reference tools that will allow you to concentrate on the *content* of your report rather than the formatting minutiae that word processors encourage.

This handbook cannot provide a complete guide to using LaTeX, but Chapter 7 and Appendix A will help you get started with using LaTeX and BibTeX for writing reports and assembling bibliographies.

You should use the `sow` document-class option for writing your Statement of Work. Section A.2 describes the document-class options provided by hmcclinic; note particularly the need for choosing a department DCO, documented in Section A.2.1.

Section A.3 describes the commands for specifying information that will appear on the cover page of your report (project title, team members, advisor, liaisons, etc.).

Appendix C covers some common issues that arise when using LaTeX and BibTeX for these tasks.

Important Note

Note that the `sow` document-class option forces the document to typeset as a report rather than as a book. Thus your Statement of Work should use the `\section` command as the top-level structural command rather than the `\chapter` commands you'll use in your midyear and final reports.

If you base your midyear and final reports on your Statement of Work document, you'll need to change the section commands to the next highest level (i.e., `\section` to `\chapter`, `\subsection` to `\section`, and so on).

5.6 Negotiating Your Final Statement of Work with Your Sponsor

Your Statement of Work is a contract. It is the document that defines what your team will produce and what your sponsor will provide to make that work possible.

The trick is to figure out how to promise the right amount of effort without promising too much—you need to commit your team to goals that are within your abilities in the available time and that will make the sponsor happy. Be as modest as is reasonable. If you deliver everything you've agreed to, or more, you will please your sponsor. If you deliver less, you will disappoint yourselves as well as your sponsor.

The Statement of Work is not a one-sided document. If your team needs anything from your sponsor to complete your project—data, software, hardware, literature, services—*anything* essential to your project that you can only get from your sponsor, you need to ask for it in your Statement of Work and make the consequences of not receiving those items clear to the sponsor.

You should be aware that your liaison may (in good faith) promise more than the sponsor's lawyers will actually allow. In particular, some types of data, software, hardware, and other proprietary or expensive items often prove not to be available. So be sure to include a last-acceptable date in the Statement of Work and an alternative course of action that you will pursue

if you don't receive the promised items by that date.

For example, failing the receipt of promised data you might state that you will generate your own simulated data. Failing the receipt of software, you might state that you will prepare alternative prototype software, but you should recognize that writing that software might turn out to be a major undertaking and affect your ability to make progress on the core problem. Thus you should take into account the fact that the sponsor's failure to deliver on their promises may slow you down and require that you moderate your commitments accordingly. Cautious contingency planning can save you grief later.

If your sponsor demands more work than you think you can reasonably commit to doing, try using phrases such as, "Time permitting, we will attempt to do X", or, "If our research leads successfully to A, we will then proceed to investigating B." Statements such as these show that you are aware of where your sponsor wants to go and that you are committed to trying to get there, but also serve as fair notice to the sponsor that you believe it may be asking too much to insist on B given the reach of the project's goals and the restraints your team is working within.

Chapter 6

Creating and Submitting a Budget

Your team may be entitled to spend money for budgeted items after you have submitted a budget to the Clinic director and it has been approved. Your budget should cover items such as travel to your sponsor's workplace, telephone calls to your sponsor, and photocopying of incidental journal articles and short reports. It should also cover any purchases made by the team; for example, books, meals, tools, and other small items.

Any unused budgeted funds will be saved and used for equipment upgrades and other programmatic costs that will benefit future students, so be sensible and frugal.

If you do need to purchase anything, your budget is due on September 24 (Mon).

Important Note

When purchasing supplies for your Clinic team (books, software, hardware, etc.) keep the following guidelines in mind:

- When possible and reasonable, have DruAnn Thomas make the purchases for you. The reasons for this are three-fold:
  - It saves you from having to submit a receipt and get reimbursed.
  - The Clinic Coordinator can see if we already own the item and avoid duplicate purchases.
  - The Clinic Coordinator can advise you about whether purchase is acceptable or not before you spend your own money on it.

- Consider the potential reuse of the item after your Clinic team is done with it. If the cost of a reusable version of the item (such as a printed book) is not substantially more than that of a nonreusable version (such as an e-book), and the item is likely to be useful to a future Clinic team (e.g., a textbook on machine learning versus a project-specific book on dry-cleaning–waste disposal), purchase the reusable version.
- The library on the third floor of Sprague includes many books that are labeled as the property of Math Clinic. Clinic teams may keep these books in their workrooms during the school year for use on their Clinic projects.

  Before purchasing new books, check to see if we already own books that meet your needs.
- For computer equipment, check with Molly Reeves and DruAnn Thomas in case we already own the item (or a suitable substitute). And see Section 6.4 on getting approval of any hardware or software purchases.

6.1 Filling Out the Estimated Budget Form

The *Estimated Budget Form* provides you with the basic skeleton of a project budget. It is available available online, at https://www.math.hmc.edu/clinic/participants/forms/.

You are required to submit an explanation for each budgeted item and a justification for each estimate on a separate sheet of paper. For example,

- If you need a specialized software package, explain what it is and why you think you will need it. If other similar but less expensive products are available, explain why you need the more expensive product. (See Sections 3.10, 6.3, and 6.4.)
- If you need travel and transportation funds, explain where you need to go, why, and how you arrived at your estimated travel expenses. For example, explain your per diem amount for food (and lodging if required). You should provide evidence that your estimates are based on competitive pricing (i.e., include estimates from multiple vendors). (See Section 6.5.)

After final approval of your budget by the Clinic director, your project manager may purchase budgeted items from those categories for which the total cumulative cost does not exceed $500, excepting consulting and consultant-related charges (all consultant-related costs require the Clinic director's approval) and computer hardware and software (which requires the approval of the department's information technology analyst).

6.2 Reimbursement

To be reimbursed for any purchase, the item must be budgeted and you must present the *original receipt* for the purchase *within 30 days of the purchase or trip* to the Clinic Coordinator, DruAnn Thomas. If you need copies of receipts for your own records, make them before submitting your reimbursement request.

Important Note

- Items that are not budgeted are not reimbursable.
- No reimbursement checks can be issued without a receipt.

6.2.1 Meals and Entertainment

Throughout the course of your project, your team will meet with your sponsor's liaisons at the college or at their site. On these occasions it may be necessary or useful to have a business meal (breakfast, lunch, or dinner). Meals are generally considered to be business-related only when the project is discussed as the primary topic of conversation. Meals while traveling should be included in your travel budget. Business meals must include both your faculty advisor(s) and your liaison(s) unless the meal takes place during an approved business trip. Exceptions must be approved by the Clinic director.

To be eligible for reimbursement as a business meal, you must submit documentation showing that a business discussion was held during the meal. Your reimbursement request must include appropriate (original) receipts including detailed food receipts listing beverages, food items, and tip information, a complete listing of attendees, and a brief description of the meeting.

The following limits apply to Clinic business meals:

Breakfast $10.00 per person attending

Lunch $15.00 per person

Dinner $25.00 per person.

NOTE: Alcoholic beverages are *not* reimbursable.

An acceptable gratuity (tip) is between 15% and 20%. If your gratuity is beyond 20% it may not be reimbursed. (Note that restaurants may impose a standard gratuity charge for groups of diners; make sure that charge is within this range.)

Your total food expense allowance is $200.00 for the project year.

Important Note

Late night "snacks" are not approved business meals under the Clinic policy, even if the team discussed the project while the snacks are consumed.

6.2.2 Items Not Eligible for Reimbursement

Harvey Mudd College has a policy for reimbursement of expenses for travel, entertainment and other business expenses. Not all items are authorized for reimbursement (such as hotel movies, rental car upgrades, traffic, parking violations, and alcohol). To view the policy and to obtain reimbursement forms, please check with the Clinic Coordinator.

6.3 High-Value Purchases (>$500)

A purchase order must be obtained for any item for which the cost exceeds $500. Written justification for each such purpose must be provided with the request for the purchase.

The Clinic director's written approval is required for any expenditure in a category for which the total budget exceeds $500 or for any consulting and consultant-related charges.

6.4 Computers, Commercial Software, and Other Equipment

More significant expenses, such as computers, commercial software, or other durable equipment that can be used by other Clinic teams or by the department, may be paid for from a different source of funds.

Purchase of computer software and hardware requires the approval of the department's information technology analyst in addition to any other required approvals to ensure that the purchases will work with our existing systems and can be reused for other projects in the future.

6.5 Use of HMC Vehicles and Automotive Rentals

Students, staff, and faculty who are registered as "authorized drivers" may use a college-owned vehicle or rent a vehicle for approved Clinic purposes. Insurance is provided by the college for these vehicles and their occupants. You will need to fill out an *Authorization to Obtain Motor Vehicle Record* form and submit the form and two copies of your driver's license to the Clinic Coordinator. This process can take up to four weeks, so you should do it early in the first semester to ensure that you are covered when you need the vehicle.

Important Note

Insurance cannot be provided for personal vehicles. *Do not use a personal vehicle for Clinic activities!*

The college's *Vehicle Insurance Policy* document,[1] maintained by the Business Affairs Office,[2] has all the details about the program, but the key points are as follows:

- If you have a purchase card issued by the college, *use it to rent the vehicle*.
- If you do not have a college-issued purchase card, purchase the optional "Collision Damage Waiver" (sometimes called a "Loss Damage Waiver") *and* make sure that the contract includes a phrase such as "HMC

[1] Available from http://www.hmc.edu/bao/forms-and-policies/.
[2] http://www.hmc.edu/bao/

business" that clearly indicates that the vehicle is being rented for business purposes.

But

- *Do not* purchase any optional liability coverage; you are covered by the college's liability insurance.
- *Do not* accept the offered full tank of gas (sometimes called the "fuel purchase option"); fill the car up yourself at the end of the rental and file the receipt for reimbursement. Clinic will not reimburse you for the rental agency's fuel-purchase charge if you accept it.

6.6 Supplemental Budget Requests

If in the course of your Clinic you find that you have underestimated your budget, you may resubmit a supplemental budget request form. This form is identical to the *Estimated Budget Form*, but your justification must include (1) a statement that your request is supplemental to a previously approved budget, and (2) an explanation for your cost overrun. Be sure to include a copy of your approved estimated budget (or budgets if there was more than one).

Remember that a supplemental budget request may be denied. Don't take chances on overspending with the expectation that you will be reimbursed.

Chapter 7

Preparing Mathematics Clinic Reports

In addition to your project's Statement of Work (see Chapter 5), you will be expected to produce two major reports during the course of the year.

The first is a midyear report, which serves several roles: a means of updating your sponsor on your results to date; an initial draft of your final report; and as a gentle introduction to the tools, skills, and outlook required when writing professional reports.

Although your midyear report serves as a draft of your final report, that document will still require a great deal of additional work, so you will need to plan ahead to avoid collision with all of the other activities (final exams, other papers and projects) that typically crowd up at the end of the year for graduating seniors.

This chapter is meant to provide you with a brief introduction to the report-writing process. You will also learn a great deal from working closely with your advisor, your liaison, and your teammates.

7.1 Reference Books on Writing and LaTeX

Writing is hard, skilled work, and LaTeX's syntax can be daunting for the newcomer. Obtaining some well-written, respected references to call on when questions arise is a worthwhile investment.

7.1.1 On Writing

We highly recommend three books for guidance on various levels of the writing process.

Lyn Dupré's *BUGS in Writing: A Guide to Debugging Your Prose* (1998) provides an excellent guide to technical writing, including motivations, terminology, formatting (including footnotes, tables, figures, and much, much more), and word choice. Dupré has edited many major works in computing and other sciences, and is acutely aware of the special needs of scientific writing.

Jean-Luc Doumont's *Trees, Maps, and Theorems: Effective Communication for Rational Minds* (2009), has of solid advice on structuring documents such as your report, creating presentations and presenting them in front of an audience, and some advice on other topics, such as effective e-mail messages and poster design.

The last book is recommended more as a reference and a stylistic tie breaker than as an immediate influence. *The Chicago Manual of Style* (University of Chicago Press, 2003) is the ultimate distillation of the stylistic rules for the University of Chicago Press, one of the oldest and largest academic presses in the United States. *Chicago* has the down-and-dirty details on every last-minute quibble about how to spell words, capitalize names and titles, number sections, lay out figures and tables, and almost anything else you can think of (and a good deal more you've probably never considered). Note that we have used *Chicago* as a guide to designing and organizing the Clinic class and sample reports.

7.1.2 On LaTeX

For LaTeX, we recommend George Grätzer's *More Math into LaTeX*[1] (2007), which is a comprehensive introduction to all of the most frequently used aspects of LaTeX, with a special—and unique—emphasis on the use of the American Mathematical Society's AMS-LaTeX packages, which are almost universally used by mathematicians in preparation of articles and books, even for journals not published by the AMS.

Grätzer's book is especially useful for its mathematical typesetting examples. He presents the typeset equations—which can be skimmed through to look for examples similar to your particular problem equation to use as guides—in conjunction with the code used to typeset the examples.

The first part of the book walks you through the process of creating your first LaTeX document. In the appendices you will find tables of useful symbols

[1] *More Math into LaTeX* is available on SpringerLink; see the bibliography for a link.

and other commands, a brief history of TeX and LaTeX, and instructions for installing a TeX system on your Mac OS X or Windows machine.

Note that the third edition of Grätzer's book, *Math into LaTeX*,[2] (2000) is also an acceptable guide.

All teams have been provided a copy of this book in their workrooms.

7.1.3 Additional References

Some other books you might want to consult include

- *Mathematics into Type*, Swanson et al. (1999).

  A guide to writing and typesetting mathematics, with a great deal of useful information on how best to compose your mathematical expressions for accurate, readable, and attractive typesetting.

- *The LaTeX Companion*, Mittelbach et al. (2004).

  Compiles the documentation for many LaTeX packages and provides code examples and typeset results to help you choose the best package for a particular use. The *Companion* also serves as an introduction to and reference for LaTeX programming, and as such can help you create your own LaTeX packages and document classes, but will also provide almost everything you're likely to need for composing your own commands, environments, and other LaTeX macros.

- *The Elements of Typographic Style*, Bringhurst (1996).

  A guide to the intricacies of typesetting, page layout, typeface design and choice, and how to format almost any textual material.

7.1.4 Obtaining References

BUGS in Writing and *Math into LaTeX* can be found in each Clinic workroom. The other books we've mentioned should be available from the Libraries of the Claremont Colleges, bookstores, and other sources.

7.2 Formatting Your Reports with LaTeX

As discussed in Chapter 5, you will be expected to use LaTeX to produce your midyear and final reports.

[2]The third edition of *Math into LaTeX* was edited by the department's system administrator, making her a useful resource as well.

The same class file used for your Statement of Work, hmcclinic, is used to provide the basic formatting of your midyear and final reports. For midyear reports, the `\documentclass` command should be used with the `midyear` document-class option, as in

```
\documentclass[midyear]{hmcclinic}
```

For final reports, drop the `midyear` document-class option; as in

```
\documentclass{hmcclinic}
```

Much more information about obtaining and using the hmcclinic document class is included in Appendix A. For more resources on using LaTeX, see Section 7.9.

7.2.1 Report Template

Included in the distribution of the hmcclinic class is a template file for use in formatting your reports. The file is called `clinic-template.tex`, and should be copied to your working directory and modified to include information relevant to your team and project.

The template file contains extensive comments explaining what the included commands do and how they should be filled out. Section A.1 covers the various commands provided by the class file for specifying the title of your report; the name of your sponsor; and the names of team members, advisors, liaisons, consultants, and so forth.

7.2.2 The Sample Clinic Report

The template file gives you the basics to get started, but we also have a sample Clinic report that can be typeset with the hmcclinic class to give you an idea of what a mostly filled-out report should look like. The sample report contains examples of most commonly used LaTeX commands and environments, along with some tips and tricks for writing LaTeX documents.

In addition to its value as a visual example, the code for the sample document is meant to serve as an example for you to emulate when writing your own LaTeX code. It contains extensive comments, and also shows best-practices layout of various LaTeX structures such as tables, figures, lists, and so forth.

The sample-report code is available from the department's website at https://www.math.hmc.edu/computing/support/tex/sample-report/ in both

`tar.gz` (compressed tape-archive files) and ZIP files. That page also includes links to the document as typeset with the hmcclinic class and the hmcthesis thesis class file, which shows how the same material can look somewhat different when typeset with a different class file.[3]

Important Note

Don't use the sample thesis/Clinic report as the basis of your document—use the provided template file instead. The sample document loads several packages that you won't need, defines some commands you probably won't use, and so forth.

Otherwise, feel free to crib from the sample report where doing so is useful, though!

7.3 Getting Started

When you're staring at a blinking cursor, the task of writing your Clinic report may seem impossible. This report may be the longest and most complex document that you've ever worked on. But like the project itself, tackling the report is a lot easier if you look at it as a series of smaller tasks that combine to create the final result.

The best way to write your report is to do so incrementally—adding elements as they become available. And, yes, that does mean that writing your report *while* you're doing your research is a (very) good way to work.

Start early in the first semester by setting up your team's report template, adding in all of the appropriate names and titles. Then, within the first two to three weeks, you can fill in the sections that describe your sponsor's organization and outline your research problem and approach. These items can be based directly on the material you write for your Statement of Work (Section 5).

As part of the process of writing your Statement of Work you will also have already begun investigating background publications in the library and on the Internet, so you can write a brief discussion of each reference while starting to compose your (annotated) bibliography. (See Section 7.7.)

[3] As you will probably notice, the differences in the documents are fairly minor, as the hmcclinic and hmcthesis class files are very similar. A more dramatic change can be seen by typesetting an article-length document with both the LaTeX article class and the amsart class created by the American Mathematical Society.

Viewed in this manner, your report is a cumulative work that is subject to evolutionary improvement at every step of the project. By working on your report throughout the course of the year, most of it will already be complete by the time that it's due, and you can concentrate on the chapters and sections that detail your research results.

Your faculty advisor can be a great resource while you're writing your reports. Keep your advisor closely involved as you work on drafts to assist with matters of style as well as content. Be sure to also solicit comments and suggestions from your liaisons, other interested faculty, and your peers.

7.4 Division of Labor and the Importance of a Single Editor

Each member of your team is likely to be particularly interested in parts of the project and not as interested or involved in other parts. For example, one team member might take the lead in devising an algorithm to solve the problem, another might write most of the code to implement and test that algorithm, and yet another might be responsible for actually running the tests, finding bugs, and fixing them. Thus it makes sense that you might have each of those team members write up the part of your team's report dealing with their particular areas of interest and work.

But every individual has their own writing style, and reading a report that has been assembled by simply concatenating the work of several people can be jarring and unpleasant. It's also possible for different team members to understand and use the same terms in slightly different ways, or use different notations for the same ideas, which can confuse your readers.

So it is very important that *one person* act as the editor of your reports. That individual should

- Assign (with consultation) different parts of the report to different team members
- Assemble the report from the submitted pieces
- Read through the report to ensure the continuity of your ideas, consistency in usage of terminology, and consistency in notation
- Rewrite the report so that it reads as though it had been written by one person

For this method of report writing to work, each team member must commit to completing and turning in their work according to the schedule that you've agreed on. Timeliness is especially important if some parts require others to have been written before they can be started. If you have such dependencies in your report, be sure that everyone involved understands the schedule and their part in completing the necessary writing.

7.5 The Structure of Your Report

LaTeX provides a series of nesting *structural commands* to specify the structure of a document. By separating and organizing related material, you can make the organization of your report clearer to your readers. A nice side effect of this approach is that it supports the division of a larger document into individual files that can be worked on independently by multiple authors.

The highest level of structural division is that of the *front matter*, *main matter*, and *back matter*, indicated within your source code by the `\frontmatter`, `\mainmatter`, and `\backmatter` commands. If you're using the template file (Section 7.2.1), these commands are already in the right places, with comments explaining what material belongs in each part.

We will provide a brief summary in the following sections; this material is drawn primarily from *The Chicago Manual of Style* (University of Chicago Press, 2003).

Important Note

Note that the order of these elements matters. If you're not using the template (see Section 7.2.1), be very sure that the order of elements in your reports matches those in the template or the sample report.

If you're not sure, ask.

7.5.1 The Front Matter

The front matter is the material that appears at the—wait for it!—front of your document! If you look at a published book, the front matter generally corresponds to the parts of the book that are numbered using roman numerals.

In the front matter of your report, you should have the following elements:

Title Page Formatted for you by the document class.

Copyright Page Formatted for you by the document class.

Abstract A *brief* (one to four paragraphs), high-level description of the main thrust of your report. You want to explain what the problem is, how you went about solving it, and perhaps have a very brief summation of your solution.

Table of Contents, List of Figures, List of Tables These elements are produced by LaTeX's automatic numbering system and their placement is specified using the commands `\tableofcontents`, `\listoffigures`, and `\listoftables`, respectively.

Important Note

The ordering of these commands is important—the table of contents should appear first, followed by an optional list of figures and an optional list of tables.

If you have other elements that you think should be similarly itemized, talk to your advisor about how to do so.

Acknowledgments A place to thank individuals, organizations, or even things that were particularly helpful in completing your project. You should also thank anyone who read your drafts and made suggestions for improving them (advisors, liaisons, others).

7.5.2 The Main Matter

As its name suggests, the main matter is where the real content of your report is found. For a report of this significance, we would expect to see several chapters dealing with the key elements of your project.

7.5.2.1 The Executive Summary

The Executive Summary is a one to two page nontechnical overview of the project and its results. Its audience is executives at the sponsoring organization who would like to quickly get a sense for your project, its results, and the value to the organization of having sponsored a Clinic project. It should give a summary of the problem statement and the solution approach (avoiding technical jargon), and, most importantly, summarize the key results and recommendations in the organizational context.

7.5.2.2 The Introduction

You should start with an introductory chapter (usually just called "Introduction") that explains the problem, gives a bit of background on your sponsor's organization, talks briefly about previous research, and then sketches out the remainder of your report with references to the in-depth discussions.

If at all possible, your introduction should be written using *nontechnical* language. Write for an intended audience of *non*–math-major HMC seniors, without assuming any specific knowledge of the area of your Clinic work.

7.5.2.3 Other Content

You should then have chapters covering topics such as

- The theoretical background of the problem, previously done related research, and how it applies to your problem
- Your theories and approach to solving the problem
- How you implemented your solution
- The result of applying your solution to the problem
- Societal impact of your work
- Other related material

See Section 5.4.1 and especially Figure 5.1 for more ideas on the sorts of things you should include in these chapters.

7.5.2.4 The Conclusion

The most important chapter in your report is the conclusion. Many readers will only read your introduction and conclusion. Some might even skip the introduction, which means that your conclusion is even more important.

Your conclusion shouldn't be just another summary of the whole process. You need to say what it is that your team accomplished over the course of the year, and why what you did was meaningful. You may want to discuss new insights you've had into the problem as a result of following the approach that you did. You might even want to include some criticisms of your work in light of what you've learned.

7.5.2.5 Future Work

You should talk about some of the possibilities for future research based on the work that you've done. For your midyear report, this section will be the place to talk about the work that is currently in progress and where you expect it to lead. For your final report, this section might grow to be its own chapter, outlining several possible future approaches to your problem and why they might be worth pursuing.

7.5.2.6 Appendices

Appendices are extra bits of information that aren't necessary for understanding the key ideas of your report, but that may be interesting for people who are especially interested in a particular topic.

Good uses for appendices are in-depth discussions of how a particular theory works, why an algorithm was constructed one way and not another, and other background material about your sponsor or a problem that may provide some additional insight into your project.

7.5.2.7 Source Code and Algorithms

You might be thinking that an appendix would be the perfect place to include the thousands of lines of MATLAB, C++, or Java code you wrote to implement a solution to your project's problem. It isn't.

By and large, people don't read printed source code. And if they are interested in your code, they probably want to run it, not just read it, and they certainly won't want to have to type it all into a computer.[4]

Source code will be included on the CD-ROM that is placed in the bound report and sent to your sponsor. It should be well commented and written so that it is easy to understand by a reasonably skilled programmer.

If you want to describe one or more algorithms in detail, especially if they are too complex to easily handle within the normal flow of your implementation and results chapters, you may do so in one or more appendices. But no source code!

More useful to most sponsors is a brief description of how to use the code, in clear, step-by-step instructions. Additionally, some sponsors find it

[4]Note that the typesetting process can often make cutting and pasting from a PDF version of your report impossible, too, as some fonts and font encodings can result in glyph substitutions that produce corrupt code when pasted into an editor.

```
\begin{figure}
  \centering
  Content Appears Here
  \caption[An Example Figure]{An Example Figure.}
  \label{fig:example-figure}
\end{figure}
```

Figure 7.1 Code for an Example Figure.

```
\begin{table}
  \centering
  \begin{tabular}{lll}
  \toprule
  Header  &  Header  &  Header \\
  \midrule
  Cell    &  Cell    &  Cell   \\
  Cell    &  Cell    &  Cell   \\
  \bottomrule
  \end{tabular}
  \caption[An Example Table]{An Example Table.}
  \label{tab:example-table}
\end{table}
```

Figure 7.2 Code for an Example Table.

useful to receive an annotated list of the code files that explains what each code file does and the format of the inputs and outputs.

7.5.2.8 Figures and Tables

We expect that you will use LaTeX's `figure` and `table` environments to format your figures or tables. The basic form of these constructs is shown in Figures 7.1 and 7.2.

Some things to notice in these examples:

Optional Float Arguments Note the lack of the optional placement argument to the float environments. In general, you should allow LaTeX to put tables where it wants to put them. If you absolutely have to have a float appear in a particular location, then you can add this argument.

Header	Header	Header
Cell	Cell	Cell
Cell	Cell	Cell

Table 7.1 An Example Table.

In general, floats should appear at the top or bottom of a page; if you have them embedded in the middle of a page, you should think seriously about whether that placement is absolutely necessary for a reader to understand your work.

It's also a good idea to leave details like the placement of floats until very near the point that you're ready to submit your final report. Until then, changing other parts of your document may cause everything to reflow and make your specification incorrect.

Optional Argument to `\caption` In both figures and tables, a caption appears with the figure or table and in the list of tables or list of figures in the front matter. Generally, the caption appearing in the list of tables or figures should not be as lengthy as that appearing in the body of your report. To create a second, shorter, caption, you should provide an optional argument to the `\caption` command. Note that here we drop the period from the required caption argument. It's especially important to use the optional argument if your caption is more than just a few words. (In general, lengthy discussions should take place in the text that references the figure rather than in the figure caption, although it's also reasonable to have a title sentence followed by one or two brief sentences describing the content of the float. Consistency is the key—if you have one or two floats with very different captioning, we'll call you on it.)

Use of the `booktabs` Package In Figure 7.2, our table uses rule commands provided by the `booktabs` package, which provide horizontal rules as shown in Table 7.1. Additional (and more complex) table examples can be seen in other parts of this document as well as the sample thesis/Clinic report.

Important Note

a. A Raster Image (72 dpi PNG image).

b. A Vector Image (PDF).

Figure 7.3 A low-resolution raster image (a) compared with a scalable vector image (b). Both are the same size, but scale very differently.

Do not use the table style with vertical rules shown in some LaTeX books.

7.5.2.9 Image Formats

You will be typesetting your report using pdfLaTeX, which only understands images in PDF, PNG, or JPEG formats.

In addition to using the accepted formats, it's important to understand the difference between *vector* and *raster* or *bitmap* images.

Vector images are encoded such that they can be rescaled with no loss of quality. Examples of vector image formats include PostScript, Encapsulated PostScript (EPS), and PDF.

Raster images are stored as pixels. They generally have a set *resolution*, often specified in *dots per inch* or *dpi*. Because of this set resolution, raster images do not scale up well. For example, a typical image on a website has a resolution of 72 or 96 dpi (screen resolution). In contrast, the standard resolution of a modern laser printer is 1200 dpi.

Scaling such an image up requires it to attempt to fill a larger space with the same number of pixels. Depending on the scaling method in use, the image will degenerate so that it either looks dotty (like an image in a newspaper) or blurry (as the program tries to fill in the gaps with guesses). Figure 7.3a shows a 72 dpi image scaled up slightly from its original size.

Raster images *can* be used for high-resolution printing, but you generally need to have a high-resolution (minimum of 300 dpi) image to use. Figure 7.3b shows a 600 dpi PDF image scaled up slightly from its original size.[5]

Finally, note that some applications will allow you to save a low-resolution raster image in a vector format. Unfortunately, that generally means that the application is creating a vector "wrapper" around the low-resolution bitmap image.

7.5.2.10 Recommended Graphics Applications

We expect that most of the images you include will be diagrams illustrating some process or graphs or charts showing your results. You may also need to include photographs that document some particular aspect of your project (or, for some projects, may actually be examples of images that are the subject of your investigations).

The following suggestions may help you find an appropriate application; feel free to consult your advisor, your liaisons, or the department's information technology analyst for additional suggestions.

Diagrams For diagrams, we strongly recommend the use of OmniGraffle Pro, a drawing application specifically designed for creating diagrams. OmniGraffle allows you to save your final images in various formats; you should generally use PDF for the best results.

You may also find the PGF and TikZ LaTeX packages useful for creating diagrams and illustrations right in your reports. This option is particularly useful if your diagram is more easily generated by a script or program rather than drawn by hand.

Graphs Many of the mathematics applications available on the department's computer systems can be used to draw graphs, either as their main focus or as one of many output options. In particular, Maple, Mathematica, MATLAB, R, and SAGE can create graphs. On the Macs, Plot and Abscissa are graphing applications. Grace and Matplotlib are Linux applications for

[5]Note that the resolution issue causes particular problems for screenshots taken of computer applications. If you need to include screenshots, you should explore the possibility of mocking up the screenshot using an application that can produce vector graphics. Alternatively, you may be able to take a screenshot of a large window and reproduce it at a smaller size, but you can still expect to see some blurring when printed.

graphing. And, of course, spreadsheet applications, such as Calc, Excel, and Pages, can generate graphs.

As with other diagrams, in general you will want to get vector output of your graphs for inclusion in LaTeX documents. In many cases you will be able to save the graphs directly as PDF, which is the best option. If you cannot generate PDF, you may be able to generate PostScript or Encapsulated PostScript (EPS), which can be converted to PDF. Avoid bitmap formats (BMP, GIF, JPEG, PNG, etc.).

Photographs For the most part, your use of photographs should be fairly basic, and may even be limited to simple color adjustments or cropping. In such cases, we strongly recommend that you use the iPhoto or Preview applications on Mac OS X, which will allow you to do these simple manipulations easily.

More complicated editing may require a dedicated bitmap editing application, such as Photoshop or the GIMP (on Linux).

For photographs, you will want to retain the highest resolution image or the most pixels that you can. Do your cropping without changing the resolution or size of the image, and adjust the image size in your document with the commands provided by the graphics and graphicx packages (Carlisle, 1999).

7.5.3 Back Matter

The back matter comprises elements such as your bibliography, an index, a glossary, a colophon, and so on. Of these, the only one you're likely to bother with is the bibliography, which we cover in Section 7.7.

If you're interested in including one or more of the other commonly seen back-matter components, please talk to your advisor and the department's system information technology analyst about the best way to do so.

7.6 The LaTeX Label and Reference System

LaTeX provides commands that allow you to refer to other parts of your document without keeping track of the numbers that it assigns to various structural elements during the typesetting process.

In addition to the numbering for chapter, sections, and similar structural elements, LaTeX also numbers mathematical equations (when using the

Object Type	Abbreviation	Reference Example
Chapters	`ch`	`Chapter~\ref{ch:`*`thing`*`}`
Sections	`sec`	`Section~\ref{sec:`*`thing`*`}`
Subsections	`sec`	`Section~\ref{sec:`*`thing`*`}`
Subsubsections	`sec`	`Section~\ref{sec:`*`thing`*`}`
Parts	`pt`	`Part~\ref{pt:`*`thing`*`}`
Figures	`fig`	`Figure~\ref{fig:`*`thing`*`}`
Tables	`tab`	`Table~\ref{tab:`*`thing`*`}`
Equations	`eq`	`Equation~\ref{eq:`*`thing`*`}`
Theorems	`thm`	`Theorem~\ref{thm:`*`thing`*`}`
Lemmas	`lem`	`Theorem~\ref{lem:`*`thing`*`}`
Definitions	`def`	`Theorem~\ref{def:`*`thing`*`}`

Table 7.2 Suggested label kinds and reference forms.

`equation` and similar environments); figures, tables, and other floats (see Section 7.5.2.8); and other elements, such as list items and pages.

By using a `\label` command, you can give yourself a textual handle that you can later use to refer to the labeled item using a `\ref` command.

For example, the text "Figure 7.1" is produced by typing

```
Figure~\ref{fig:example-figure-code}
```

In the `figure` environment for that figure, we have a `\caption` command, and at the end of the command (or on the following line without whitespace between them), we have a `\label` command that looks like `\label{fig:example-figure-code}`.

7.6.1 Formatting Labels

In general, we recommend that you format your labels in a consistent, but generic, way, usually in the form *`kind:label text`*. Using `fig` for figures, `tab` for tables, `ch` for chapters, `sec` for sections, `eq` for equations, and so on, means that you can easily distinguish between labels (and, perhaps, have the same label text for both a section or equation and an illustration showing a visual example of what you describe). Some text editors take advantage of these "kind" labels to help you find and insert references, and allow you to limit your search by reference type. There are even LaTeX packages that can automatically fill in the labels for you, formatted correctly.

Table 7.2 shows some suggested label type abbreviations and reference examples. Note that for sectional commands beneath the chapter level, we recommend simply using the `sec` type and the `Section` label in references; doing so allows you to change the level of a sectional element without having to worry about fixing the label and all the `\ref` commands throughout your document.

7.6.2 Nonbreaking Spaces and References

Notice that all the examples in Table 7.2 have a tilde character (~) connecting the label to the `\ref` command. That tilde is called a *tie*, and it acts as a nonbreaking space to prevent the label from being separated from the reference (e.g., at a line or page break). You should almost always use a tie with your `\ref` commands.

7.6.3 Page References

You can also refer to the page on which a `\label`ed element appears by using the `\pageref` command with the same argument you would give to a `\ref` command. For example, Table 7.2 appears on page 74.

7.6.4 Equation References

For your thesis, we require you to use references of the form `Equation~\ref{eq:`*`thing`*`}`. In other publications, you may be required (or want) to use an abbreviation instead of the full word, or even drop the label altogether.

The amsmath package includes a `\eqref` command for formatting equation references. This command formats equation references as a number surrounded by parentheses, as in "(1.3)". In the thesis class, this command has been overwritten so that it produces our desired label form.

7.6.5 Multiple References

If you're referring to more than one `\label`ed item, you don't need to label each `\ref` command. For two, use a tie to connect the first number to the label followed by the word "and" and the next reference. For example,

```
Equations~\ref{eq:thing1} and \ref{eq:thing2}
```

For a range of items, say several figures illustrating some point, you might do

```
Figures~\ref{eq:thing1}--\ref{eq:thing4}
```

where you're referring to `thing1`–`thing4`, and would end up with something like

Figures 1.3–1.7. . . .

If you don't have a continuous range, write something like

```
Figures~\ref{eq:thing1}, \ref{eq:thing3},
and \ref{eq:thing4}
```

which would give you something like

Figures 1.3, 1.5, and 1.6. . . .

7.7 Bibliographies and Citations

Formatting bibliographies and maintaining citations is yet another difficult task that LaTeX can help you handle. The BibTeX system provides a (command-line) tool, a language to specify the structure of bibliography entries (used in `bst`, bibliography-style, files), and a format for creating a database of bibliography entries. Combined with the `natbib` package (see Section 7.7.2), BibTeX allows you to create flexible citations whose format can be changed as easily as the layout of the document.

BibTeX and `natbib` are reasonably complex; we won't cover their details here. Instead, see the rest of this section and the articles "Managing Citations and Your Bibliography with BibTeX" (Fenn, 2006) and "Tame the BeaST: The B to X of BibTeX" (Markey, 2005), as well as the appropriate chapters of any LaTeX references you may have to hand (such as Chapter 16 of Grätzer's *More Math into LaTeX* or Chapter 10 in his *Math into LaTeX*).

7.7.1 The Author-Date Style

The department uses an author-date style for citations and bibliographies. The author-date style provides more information about references than other styles, such as bracketed numerals, and is thus more accessible for a casual reader.

Examples of author-date style citations can be found in this document as well as the sample thesis/Clinic report (see Section 7.2.2).

Form	Command	Result
Compact author lists	`\citeauthor{MathWrite}`	Knuth et al.
Full author lists	`\citeauthor*{MathWrite}`	Knuth, Larrabee, and Roberts
Just publication year	`\citeyear{gratzer-mmil}`	2007
Year with parentheses	`\citeyearpar{gratzer-mmil}`	(2007)
Author names, but with years in parentheses	`\citet{gratzer-mmil}`	Grätzer (2007)
Whole citation in parentheses	`\citep{gratzer-mmil}`	(Grätzer, 2007)

Table 7.3 Examples of natbib Commands.

7.7.2 The natbib Package

The natbib package, written by Patrick Daly, supports a wide variety of bibliographic citation formats, and also provides support for leveraging the power of LaTeX and BibTeX's substitutions to allow you to place text inside citations or create parenthesized comments using a single command, such as "(see Grätzer, 2007: chapter 10)", produced by

```
\citep[see][chapter~10]{gratzer-mmil}
```

The natbib commands also allow you to insert references as names, years, or other forms. Table 7.3 shows several variants. Note that the most common usages are `\citep` or `\citet`; if you're converting an existing document that simply used `\cite`, you can often use

```
\let\cite\citep
```

in the preamble of your document to convert all the citations to use the `\citep` form, and then adjust the citations that need special treatment. (Obviously you can achieve the same thing by doing a global search and replace to change the citation commands throughout your document.)

Documentation for natbib can be found in the files `natbib.`*`ext`* and `natnotes.`*`ext`*, with *`ext`* being dependent on the exact form of the documentation on your system; some systems use DVI, some PostScript, and some PDF formats for documentation. On Unix-like systems such as Linux or Mac OS X, you may be able to easily access this documentation using the `texdoc` command; as in `texdoc natbib` or `texdoc natnotes`.

`natnotes` is a handy "cheatsheet" for the various commands provided by the natbib package.

7.7.3 The hmcmath Bibliography Styles

The format of your citations and bibliography are provided by the bibliography style file. The department provides three bibliography styles for use with theses or Clinic reports:

hmcmath.bst A basic bibliographic style modelled after the style recommended by the fifteenth edition of *The Chicago Manual of Style* (University of Chicago Press, 2003).

hmcmathlabelledannote.bst The same style, but with annotations (recorded in the `annote` field) turned on.

hmcmathannote.bst The same style with the `key` field and the label for the `annote` field removed.

In most cases, you should use hmcmathannote when you're asked for an annotated bibliography, and hmcmath when you don't need annotations.

One of the key features of these bibliography styles over some of the standard BibTeX styles is their support for URLs, which are increasingly common in today's Web-oriented research community.

These bibliography style files are available on the department's Linux systems and can be used on department machines without any additional work on your part.

If you want to use these styles on your own system, or on a system not managed by the math department, they are stored in `/shared/local/texlive/texmf-local/bibtex/bst/hmcmath` and can be copied from there or downloaded from https://www.math.hmc.edu/computing/support/tex/bibliography-styles/.

7.7.4 Constructing Your Bibliographic Database

Once you've read some of the references suggested in Section 7.7, you should be able to start creating bibliographic databases for use with your report.

7.7.4.1 BibTeX Tools

There are many tools that can make composing BibTeX bibliography databases easier. The MacTeX TeX system we recommend for use on Macs includes

BibDesk, a graphical bibliography tool. Windows tools include BibDB. A cross-platform (Java) tool is JabRef.

A Google search will help you find more tools, but we would also encourage you to consider another cross-platform solution: Emacs.

Emacs (GNU Emacs and XEmacs) has excellent support for bibliography databases with `bibtex-mode`, which is included with Emacs. It helps you enter new records, clean up existing records, validate records, sort records, and can even convert some non-BIBTEX records to BIBTEX format. Combined with AucTeX and RefTeX, Emacs provides a complete editing environment that allows you to compile your document, track changes, interact with version-control systems (see Section 4.8), navigate your multifile documents, and insert and track `\label`, `\ref`, and `\cite` commands.

7.7.4.2 BibTeX Tips

A few less-than-obvious tips that might come in handy include

- Some of the web-based bibliographic databases include the option to export citation information in BibTeX format. Sometimes these citations are everything you need, sometimes they need some additional work to make them as useful as possible, but they're usually a good start. Such BibTeX records may even include annotations, notes, or other useful information (although you should create your own annotations if you're producing an annotated bibliography).

- Choose meaningful citation keys (and be consistent!). Citation keys are the strings that you use to refer to bibliographic records in your LaTeX document. Your keys should give you enough information that you can identify—at least roughly—the source from the key alone when reading through your LaTeX code. Many of the searchable databases have keys based on some obscure coding, such as a DOI[6] or the Mathematics Subject Classification. These identifiers work great within the system they're part of, but they're meaningless to most human beings.

  Instead, choose keys that *you* will recognize on sight; for example, `smith-1990-english-cocoa` for an article by John Smith, published in 1990, about the varieties of cocoa available in the English market; or `turing-computable` for Alan Turing's "On Computable Numbers, with an Application to the *Entscheidungsproblem*" (1936).

- When entering names, use the *`last name`*`,`*`first name`* format for all the names, separating multiple authors' names with `and`. Being consistent about this data-entry method makes it easier for you to avoid mistakes with extra commands, reversed names, and so forth. It can also help BibTeX deal with some issues with sorting names with multiple particles (e.g, "Per Brinch Hansen", whose last name is "Brinch Hansen"; or "Charles Louis Xavier Joseph de la Vallée Poussin", whose name should be entered as `de la Vall\'{e}e Poussin, Charles Louis Xavier Joseph` (Walsh, 2002)).

  This method can also help you be clear about the correct order of

[6] Digital Object Identifier, a globally unique identification system meant to label digital content. See http://www.doi.org/.

names in languages that place the family name first, such as Hungarian and many Asian languages.

- Try to find the most complete name you can for an author. If a BibTeX record you find online only has "A.N. Author" (or "Author, A.N.", as we recommend here), try to find out the author's first name. (Middle names are a bonus.) When a last name is common, there could be several first names that abbreviate down to the same initials; distinguishing between "Sally Smith", "Samantha Smith", "Samuel Smith", and "Steven Smith" when all *you* have is "S. Smith" is not an easy task.

  Note that the formatting of the author names in the bibliography is handled by the bibliographic style you choose. So one journal might want full names, and another abbreviations, but BibTeX will handle the formatting for you.

  It's always better to have more information than you need and throw some of it away than to not have enough.

- Enter titles using title caps; that is, rather than entering, "An article about important topics"; enter "An Article About Important Topics". Here, again, BibTeX handles the formatting, and it will downcase words when needed. But BibTeX can't add capitals.

- Protect required capitals within entries. Sometimes you have proper names, countries, acronyms, or other strings that should always be capitalized, even if the rest of the string is downcased. Stop BibTeX from changing them by enclosing the capital letters in braces. Examples include

```
An Exploration of {C}harleston's {T}heorem
{S}wedish Mathematics in {N}orwegian Classrooms
How the {U}nited {N}ations Formed {UNICEF}
On Computable Numbers, with an Application to
  the \emph{{E}ntscheidungsproblem}
```

7.7.5 Citing HMC Theses and Clinic Reports

You might find yourself wanting to cite a previous student's thesis or Clinic report. BibTeX provides a number of different possible ways of doing so, but we suggest the following:

Theses Cite theses with the `@Misc` entry type. Use the `title`, `author`, and `year` from the thesis, and use

> Harvey Mudd College Mathematics Senior Thesis

as the content for a `howpublished` field.

Clinic Reports Cite past Clinic reports with the `@TechReport` entry type. Fill in the `title` and `year` from the report, and use the Clinic team members' names (in alphabetical order) for the `author` field. Specify

> Harvey Mudd College Mathematics Clinic

as the `institution`, and, if the sponsor's name isn't mentioned in the title of the report, add a `note` field with something similar to

> Project sponsored by *`Sponsoring Corporation`*.

7.7.6 Citing Webpages

More and more useful material is appearing on websites every year. Where this material is a clearly defined "article" or "technical report"—especially when it's a PDF offprint that you can download and view—you can (and should) use the appropriate BibTeX entry type. But sometimes the material really is just a page on a website. For those cases,

1. Use the `@misc` entry type.
2. Set the `howpublished` field to "online; last viewed *`date`*".
3. Set the `url` field to the URL for the webpage.

If possible, you should also set the `year` and `month` fields to the publication date of the webpage (e.g., the copyright year or the date in the header of the webpage's HTML code).

7.7.7 Citing Other Electronic Documents

The `hmcmath` and `hmcmathannote` bibliography styles also support an `eprint` (for, e.g., arxiv.org prints) and `doi` (Digital Object Identifier) field, both of which are currently treated the same as a `url` field (i.e., set as a URL and made into a hyperlink).

7.8 Organizing LaTeX Source Code

For a report of this size, we generally recommend that you have

- A *master file* that contains the identifying information for your report, along with all command and environment definitions, package loading, and other code that should be available throughout the report
- *Separate chapter files* containing the material for each chapter

And, of course,

- A *separate bibliography database* file

Although the separate files approach is often seen as an advantage for groups of collaborators, as each writer can work on a separate piece of the larger document in its own file, this approach also has some advantages for people working on their own.

The organizational benefits are obvious, but if you're constantly retypesetting your document while you're trying different formatting approaches to a particularly troublesome structure or equation, you can actually save time by not typesetting the entire document each time. The `\includeonly` command allows you to decide which `\included` files are typeset. After typesetting your whole document a couple of times (to populate the auxiliary files with labels, entries for the tables of contents and lists of figures and tables, etc.), you can add an `\includeonly` command in the preamble of your document and only typeset the file you're working on, which will just give you the content of your chosen files, but with the cross-references still filled in. (By listing all your `\included` files in the `\includeonly` command and commenting out the files you're not working on, you can easily switch files on and off without having to make other changes.)

While you can also split the content of each chapter into multiple files (e.g., one for each section), it's usually easier to keep the level of chunking at the chapter level. But if your particular needs make smaller chunks more useful, talk to your advisor or the department's information technology analyst about whether additional splits are a good approach.

7.9 Getting More Help with LaTeX

At first glance, LaTeX may seem incredibly baroque and difficult to understand. After some experience (and judicious reading of documentation!), you should

be able to do most basic tasks in LaTeX without having to think about them too much.

But there are always more complex tasks that you'll know you should be able to do, but whose exact coding won't be obvious. For those situations, consult references (the examples in Chapter 5 of *Math Into LaTeX* (Grätzer, 2000) or Chapter 9 of *More Math Into LaTeX* (Grätzer, 2007) are particularly useful for complex math displays), and consult (in no particular order) the department's TeX support pages, your advisor, the department's system administrator, Google, or the `latex-l` list hosted by the college. (See https://www.math.hmc.edu/computing/support/mailinglists/ for more information on subscribing.)

7.10 Vetting and Submission of Your Report

Before your midyear or final reports are submitted to your sponsor, they will undergo a rigorous review process by your advisor and the Clinic director. You may be asked to revise your report as a result of any or all of these review steps.

Note that you can seek out comments on your draft at any time, and doing so earlier rather than later can help streamline the process.

A quick overview of the process is as follows:

1. Submit your report to your peer reviewers who will review it and return it to you with suggested changes.
2. Submit your peer-reviewed and revised report draft to your faculty advisor, who will review it and return it to you with any additional needed changes.
3. When your team and your advisor are happy with the draft, submit it to the Clinic director. The director's review is primarily to ensure that the report will make sense to an nonexpert and conforms to the department's standards for documentation.
4. After making any revisions requested by the Clinic director, your team will revise the document again, submitting your drafts to your advisor for review.
5. When you, your teammates, your advisor, and the Clinic director are all happy with the final draft, you will submit both printed and electronic copies to the department's Clinic administrative assistant.

6. For final reports, you will also be expected to create several copies of a CD-ROM containing your deliverables in electronic form. These CD-ROMs will be included in the bound reports that the department will send to your sponsor. See Chapter 9 for more details on this process.

Because the results of a Clinic project are products for which the college has ultimate responsibility, your faculty advisor and the Clinic director must retain editorial authority over the materials that your team presents to your sponsor.

7.10.1 Checking Your Code

Appendix C includes a list of frequently seen problems to avoid, which will give you an idea of the sorts of things that the code/format review will be looking at.

Running the LaCheck program on your code[7] will help you find and fix many (but not all) of the problems discussed in Appendix C. It reads in your entire document (including all the subfiles), and gives you feedback about the code. LaCheck is particularly helpful for finding issues with spacing (use or nonuse of the tie command; extra spacing in environments or command arguments; missing `\@` before sentence-ending punctuation where a word ends with a capital letter); issues with ellipses; italic corrections; correct use of quotation marks; and so on.

Note that LaCheck is not perfect, and that it may complain about things that, while problematic in the general case, may, in your particular usage, be correct. So you will need to look at each warning in context and decide whether the "problem" needs to be addressed or not.

LaCheck is available on any departmental Mac OS X, Linux, or Windows system running the TEX Live TEX system.

[7] With `lacheck` *`master-file`*`.tex`.

Chapter 8

Presentations and Posters

A major component of the Clinic experience is training and practice in giving good talks, an art form that requires practice and discipline to master.

Practice sessions are held from the start of the semester, when the material you'll be presenting is relatively simple, with the presentations becoming more complex and demanding as the project accomplishes more work and there's much more to say within a limited time.

Your talks will be somewhat informal during the fall semester, when the mathematics Clinic practices separately from the other departmental Clinics. But later in the spring semester, all of the college's Clinic teams will meet together for formal presentations. These formal talks can be seen as the warm-ups for Projects Day, May 7 (Tue), the culmination of the year's work, when you will be giving presentations not only to your fellow students and their advisors, but also to representatives of your own and other Clinic projects' sponsors, members of the general public, and maybe even your own friends and family!

8.1 Your Presentation

Remember that your presentations should be aimed at a general audience, or, at most, an audience that is only vaguely familiar with the ideas behind your work. Avoid technical jargon and excessive detail unless you absolutely must include such details to make your project's goals and accomplishments clear. It's better to give a good general overview and leave the details for the question period or for other discussions than to leave your audience in stunned incomprehension or utter boredom.

You should work closely with your advisor when preparing your talks. Your advisor should review your slides, and, if possible, attend your practice presentations and provide feedback. You may also want to get other people—friends, other thesis or Clinic students, innocent bystanders—to watch your presentation and give you feedback.

It's also important for your project manager to keep your liaison(s) informed about the content of your slides and talk. Before giving a presentation, your should ask your liaison(s) to review your slides. Keeping your liaison(s) in the loop will ensure that your final presentation is what they are expecting and doesn't violate any intellectual property agreements you or the college may have been asked to sign.

We recommend you consult Doumont (2009) for details about constructing and giving presentations.

8.1.1 Content

Your presentation should have several major parts, each split into smaller parts, perhaps represented by separate slides. We encourage you to maintain the same format for all the talks that you will give during the year. Doing so allows you to become comfortable with your presentation and to fine-tune it as your work evolves.

Introduction Begin with an introduction. Include

- A title slide, including sponsor identification and the names of all participants
- A *brief* and *relevant* overview of your sponsor's areas of interest as related to the project
- A brief description or explanation of the problem you're working on

The Heart of Your Talk Include

- Your approach, in descriptive, *nonspecialist* language
- What you've accomplished—discuss your analysis and results

Conclusion Include

- A discussion of work remaining to be done or negative results
- Recommendations for future research

Questions and Answers Open the floor to the audience. You may want to end your talk with a slide that makes it clear that you're welcoming questions.

Slides Covering Additional Material You may want to include some slides that cover some of the more advanced material that you didn't include in your talk, but that you think people might ask questions about. Although doing so is optional, actually having slides covering those questions can really wow an audience. (Hint: Figuring out what questions you might want to cover is best done by giving practice presentations and paying attention to the questions that people ask you during the practice sessions!)

8.2 Slide Content

Slides should act as reinforcements or illustrations of what you're saying rather than as a distraction. *You* are the main show; you want your audience to focus on you and what you're saying rather than having them struggle to read large swaths of tiny text on a slide, reading all your bullet points and getting ahead of where you are in your talk, or being dazzled by brightly colored, clashing borders from your slide style.

Your slides act as supplements, illustrating complex ideas or allowing you to summarize information with a graph. See Edward Tufte's essay, *The Cognitive Style of PowerPoint: Pitching Out Corrupts Within* (2006b) for more details on the negative impact of "PowerPoint-style presentations" on communicating information.[1]

Tufte's other books, notably *The Visual Display of Quantitative Information* (1983) and *Beautiful Evidence* (2006a), include excellent examples and guidelines on constructing powerful, *meaningful* documents, posters, or presentations to communicate scientific information.

Many of Colin Purrington's guidelines for designing posters (2016) are also relevant when designing talks.

8.2.1 Textual Material

Keep the textual content of your slides brief and to the point. One increasingly popular approach—developed in response to the text-rich but information-

[1] Tufte's website features a discussion thread about this essay, with posters contributing stories from their own lives.

poor style described by Tufte (2006b)—is to limit the content of slides to just a few words. Some presenters limit the text on a slide to a single word.

Stanford law professor Larry Lessig is a pioneer in using this style of presentation. Garr Reynolds discusses Lessig's style and provides links to some examples (2005). Reynolds's site, Presentation Zen, also describes many other presentation styles—if you don't like Lessig's approach, you may find another that you're comfortable with using for your own talks there.

8.2.2 Presentation Graphics

The key with graphics is to *keep it sane*. It's easy to get caught up in the idea of providing lots of shiny pictures for your audience to look at, but it's far better to keep graphics to a minimum, which increases their impact on the audience dramatically—if all they've seen so far is sparse text, seeing an image tells them that there's something important happening.

In most cases, you can describe everything you need to talk about in words. But if the best way to get a point across is to use a graphic—a graph, a diagram, a chart, even an animation—do it.

8.3 Feedback and Critiquing Presentations

The talks you give in class are meant to encourage critiques on your presentations from your peers. We encourage (nay, *require*) you to attend these sessions and to think of good questions that you can ask during the Q&A session at the end of each presentation you see. By asking (and being asked) good questions, you will learn about things that you may have left out of, underemphasized, or glossed over in your presentation. This information will help you adjust your presentation before you make it again.

If you have a sudden flash of insight into someone's problem, share it! You may help your classmate find a new way of exploring their problem and help them get better results than they might have done without your question.

8.4 Creating Research Posters

Most of the issues we've discussed about creating and giving effective presentations are also important when creating a poster—with the additional

need for the poster to stand on its own when you're not there to answer questions.

Research posters have to balance the main goal of communicating your work with that of having an attractive poster that encourages people to take a closer look. In particular, while you might be tempted to restate everything in your report on your poster, you need to keep the font size large enough that people can read it from several feet away, but there are many other things to consider as well.

Our LaTeX poster class, hmcposter,[2] takes care of many of the basic decisions you might otherwise need to make, and also happens to produce posters that we can easily print.

The hmcposter class has some disagreements with Colin Purrington's "Designing Conference Posters" (Purrington, 2016)[3], but Purrington's article is an excellent discussion of how to make a good poster.

8.5 Good Resources for Presentation and Poster Design

In addition to Purrington's articles and examples, there are a number of other good resources that are worth checking out.

- *Trees, Maps and Theorems: Effective Communication for Rational Minds*] by Jean-Luc Doumont (Doumont, 2009) is a fantastic book that covers writing, presentations, graphic displays, and more.[4]
- Garr Reynolds's web article, "Presentation Zen: The 'Lessig Method' of Presentation" Reynolds (2005), gives an inspirational argument for a minimalist presentation style, with the emphasis kept on the speaker(s) rather than the slides, using the slides as visual background or for specific illustrations rather than including much of the text of the talk on the slides themselves. (See also his more detailed book of the same title, published in 2011.)

[2] https://www.math.hmc.edu/computing/support/tex/classes/hmcposter/

[3] Article: http://colinpurrington.com/tips/poster-design. "Good" example: http://colinpurrington.com/wp-content/uploads/2011/09/scientific-poster-advice-purrington.pdf. "Bad" example: http://colinpurrington.com/2012/example-of-bad-scientific-poster/.

[4] There are also some videos and other materials viewable on line at http://www.principiae.be/X0302.php.

- Edward Tufte's books on design and data visualization, *Beautiful Evidence* (Tufte, 2006a), *The Visual Display of Quantitative Information* (Tufte, 1983), *Envisioning Information* (Tufte, 1990), *Visual Explanations: Images and Quantities, Evidence and Narrative*, and his essay "The Cognitive Style of PowerPoint: Pitching Out Corrupts Within" (Tufte, 2006b), are good in so many ways, with fantastic examples from the history of visualization and strong advice.

8.6 Dress Codes for Presentations

Your fall semester presentation is informal, and your regular clothing will be acceptable, though you should avoid clothing that is distracting.

For formal presentations, however, we may require you to wear "business-casual" clothing, which basically means

- Neat and clean appearance
- For men, long-sleeve shirts, clean slacks, and dress shoes
- For women, formal skirts or slacks and blouses, sensible shoes (i.e., low heels or dressy but subdued flats)

For Projects Day talks or poster sessions (May 7 (Tue)), we will expect more formal office-appropriate clothing:

- For men, coat and tie
- For women, suits or formal business dresses

8.7 A Last Word: Sensible Paranoia

Don't tempt fate by incorporating new, untested graphics between your penultimate and your final presentations.

And *always*, even during practice sessions, check your audio-visual equipment before your presentation and be prepared for a last-minute disaster by keeping your presentation on a USB key, accessible from a website, or even having a set of overhead slides handy, just in case.

Chapter 9

Completing Your Clinic Project

There are a number of final tasks that must be completed so that you can receive a grade for your Clinic work and be eligible for department awards, and so that we can have archives of your work and final versions of your report and other materials to mail to your project's sponsor. This chapter explains what those tasks are, how to perform them, and where to go for more information.

9.1 Due Dates

The exact due dates for the components of your Clinic project are specified in the calendar on page i of this document. If you have questions, ask your advisor or the Clinic director.

9.2 Returning Books, Software, Hardware, and Other Materials

You must return any and all books, software, hardware, and other materials that were purchased by your team with Clinic funds; borrowed from the library, faculty advisors, or other sources; or that were loaned to you by your project's sponsor.

Items to be especially aware of include

- Reference materials or other books
- Computer hardware, including actual computers, monitors, keyboards,

mice, external hard drives or flash drives, presentation controllers, projectors, laser pointers, and any other equipment

- Computer software packages used for the project, including boxes, manuals, CD-ROMs or other media
- Media (tapes, CD-ROMs, DVDs, etc.) containing data used by the project
- Keys, pass cards, or other access devices

9.3 Receipts and Final Accounting

You must submit all receipts for food, travel, or other expenses that you accrued during the course of the year for reimbursement.

Important Note

- We need the *original receipts*; copies are not acceptable.
- If you need or want to retain a copy of your receipt, you will need to make a photocopy for yourself before turning in the receipt.
- Expenses submitted without original receipts will not be reimbursed.
- Expenses submitted more than 30 days after the purchase was made or the trip was completed may be classified as taxable income and have taxes withheld.

9.4 Your Clinic Report

You will need to submit several printed copies of your final, advisor- and Clinic–director-approved report. These reports must be

- Printed duplex (on both sides of the page)
- Conform to the expected appearance of the Clinic report, as defined by the hmcclinic document class (see Appendices A and C as well as Chapter 7 for details)

- Be printed on the proper paper—we don't use standard printer paper for reports; see DruAnn Thomas to obtain the correct paper
- Be unbound, unstapled, and not hole-punched

The number of printed copies required equals the total number of students, faculty advisors and liaisons on the team, plus two additional copies for Clinic archives. DruAnn Thomas will inform you of the exact number that you need to print.

9.5 Archival CD-ROMs

Because your sponsor is the legal owner of all intellectual property generated by your Clinic work, they must be given an archive of all reports, documentation, code, and data that they would need to reproduce or continue the work. For each copy of your report (see Section 9.4), you will need to submit a CD-ROM containing computer-generated materials. These CD-ROMs will be included with each report. If your materials are large enough that they cannot fit on a single CD-ROM, you may need to submit a DVD-ROM instead.

The media you create should be readable on *all three platforms* that we support—Windows, Linux, and Mac OS X. Note that by default, newer Windows systems will create nonstandard formats that will not be readable on the other systems. Appendix D includes detailed instructions for creating CD-ROMs.

We expect that you will clean up the materials that you submit before your submit them. For example, don't include backup files that you or an editor created or the log and other auxiliary files created by LaTeX during the typesetting process.

Important Note

Do not create your CD-ROMs or DVD-ROMs until *after* the final version of your report has been approved by both your advisor and the Clinic director, and you have made all the changes they recommend.

9.5.1 Required Contents

The contents of your CD-ROM or DVD-ROM should include, at a minimum,

- Your midyear and final reports, including:
  - An electronic (PDF) copy of the compiled reports
  - LaTeX source code (excluding auxiliary and backup files generated by LaTeX
  - All original figures
- Copies of your final presentation slides (as PDF, PowerPoint, Keynote, or other format; source code for the slides is optional but recommended)
- A copy of your Statement of Work
- A copy of your final poster
- Any source code that you wrote as part of the project (see Sections D.4 and D.5 for details on extracting your code from your Git repository)
- Documentation for your code that is not included in your report
- Executables generated by your code so that the sponsor can immediately use the product if desired

---

Important Note

This list may be revised during the year.

9.5.2 Optional Contents

Optional, but desirable, contents for your CD-ROM or DVD-ROM include

- Notes, white papers, or other materials that you developed during the course of the project for your own use (e.g., contents of a wiki, quick-reference guides for using some software)
- Slides or other materials from presentations other than your final presentation
- The electronic file (or source code) used to create your poster for Projects Day
- Test data or other materials used in developing your code (but see Section 9.5.3)

- Other materials that you think would be relevant and interesting for a potential reader of your report

Even if you do not include a dump of your repository, you should include an "exported" copy of the final version of your code (i.e., a copy of the final code without any version-control metadata included).

Important Note

This list may be revised during the year.

9.5.3 Data

Data is a special case. In most cases, any data that you generate during the course of the project will be too large to fit on a CD-ROM or, in some cases, on a DVD-ROM. Depending on the nature of the data and the difficulty in reconstructing it, you might want to just include information on how the data was produced.

If the data is vital for the sponsor's acceptance of the report (e.g., they're planning on using your results as-is) and it will not fit on simple optical media, we will need to consider the best way to get the data to your sponsor. You will need to let your advisor, the Clinic director, and the department's system administrator know about this requirement in advance (preferably near the beginning of the spring semester at the latest).

9.6 Departmental Electronic Archive

The department maintains an electronic archive of information related to previous Clinic projects. This information is stored in the `/home/clinic` directory for each team on the department's file server (see Section 3.6.1 for more information on the location of this directory).

The contents of our archive should include *both the required and optional items* listed in the description of the CD-ROMs for your reports. You should include a directory, `cdrom`, in your `/home/clinic` directory that contains everything on the CD-ROM to be included in your report, but you should also include any other materials you think might be useful for a team following in your footsteps somewhere in your `/home/clinic` directory.

In particular, if you used a revision-control system (Section 4.8), you *must* include a copy of your repository in your electronic archive, although

if you're not supplying the repository dump to your sponsor, that task can be completed after the rest of your deliverables have been completed.

Just as with the CD-ROMs, other data may present a problem. Consult with your advisor, the Clinic director, and the department's information technology analyst to determine whether your data must be preserved and what the best way of doing so will be.

9.7 Contact Information

We will need postgraduation contact information for all the members of your team so that we can contact you to follow up on missing materials, get answers to questions from your project's sponsor, and, of course, send you your bound copy of your team's report.

Submit that information to DruAnn Thomas as soon as you know it. You can also send it to her at druann_thomas@hmc.edu when you know it, or send her updates if your contact information changes.

Chapter 10

Grading

Clinic projects are equivalent to a two-semester, six-unit course. During the regular academic year, you will receive a separate grade for each semester's work.

Your faculty advisor assigns your semester and year-end grades as in any regular three-unit course, so individuals on the same team may receive different grades. Your faculty advisor may also choose to write a letter of special commendation for any students who have done particularly outstanding work.

This chapter includes some information about the grading criteria (measures of performance) that may be used by your advisor in assigning course grades; you may also read this chapter for useful tips on team participation and project management.

10.1 Some Measures of Performance Used in Assigning Course Grades

Your faculty advisor will use these criteria to help them determine your semester grades. These guidelines are merely guidelines, however, and your advisor may choose to add, ignore, or weight particular criteria differently than other Clinic project advisors within the department or the college. Feel free to discuss these criteria with your advisor if you have any particular concerns or suggestions.

A Note on "Citizenship" As Clinic projects involve teamwork to a much greater extent than almost any other coursework at the college, social and

"good citizenship" factors will be taken into account in determining your grade—individual performance is not sufficient. You may, therefore, find it particularly useful to review these guidelines to help you understand how you can perform your best in your role as a Clinic team member.

10.1.1 Technical Contributors

All student participants are technical contributors to the project's success. Particular areas to consider include

- Adequacy and appropriateness of theory, library research, modeling, analysis, simulations, data analysis, computer work, conclusions and documentation
- Cooperation as a team member, taking obligations seriously and meeting deadlines
- Quality of written, graphic, and oral work
- Initiative and imagination in taking responsibility for technical work
- On-time attendance and participation in team meetings
- Attendance at the mandatory 11:00 A.M. Tuesday presentation meetings
- The content, timeliness, and other information recorded in a journal or research notebook (see Appendix E; Purrington (2011)) may provide an important indicator of the quality of your work and progress

10.1.2 Project Manager

The Project Manager has additional responsibilities over and above their role as a technical contributor. For Project Managers, advisors should also consider

- Initiative and imagination in taking responsibility for overall team performance
- Goal setting
- Consensus building
- Scheduling and maintaining work on schedule

- Achieving an appropriate and balanced division of labor among the team members
- Communication with team members, advisor, liaison, and consultants

10.1.3 About Performance

Students are sometimes unclear about the expectations for team research and how an individual's performance may be evaluated. Although different advisors may emphasize different aspects of performance, the following comments should give you a better idea of what is expected in successful completion of a Clinic project. After reading this section, talk with your advisor about their expectations so that you understand what they are looking for and can ensure that you meet those expectations.

Remember that Clinic projects are characterized by teamwork. Your performance as an individual is important, but the overall performance of the team is also important—possibly even *more* important than your individual performance. Your goal is to help your team succeed while simultaneously gaining as much useful experience in research as you can.

Your advisor may periodically poll your team's members to get your rankings of your own performance as well as the performance of your teammates. For additional perspective, a faculty advisor may ask to review your journal or research notebook on a regular basis to evaluate your progress and performance.

The following points touch on many of the aspects of performance that we are evaluating and offer some perspectives on how these criteria may be considered in assigning grades.

Problem Definition, Model Design, and Analysis

Mathematical research in solving real-world problems involves several interlocking intellectual tasks. A typical problem-solving experience might include steps such as: formulating the problem; designing a model of the problem; understanding the theory that bears on the model; using the theory or simulations (perhaps relying on computers) to analyze the model; comprehending the results of the analysis; and articulating the results into useful conclusions.

An outstanding student will deal imaginatively and proactively with formulating, modeling, analyzing, and stating conclusions.

Report Writing and Oral Presentations

If you have something worth saying, it's worth saying well. As clear as your conclusions are to you within the privacy of your own mind, they remain secrets until you make them known to others.

An important aspect of the Clinic experience is the public presentation of the results of your work. Conveying complex ideas to others in lucid and persuasive ways is a craft (an art when done superbly) distinct from discovery and development, but ultimately just as important and difficult as making the discovery itself. You will be required to both document your results in writing and make oral presentations about your team's accomplishments.

An outstanding student will demonstrate understanding of the special importance of documentation and presentation in scientific work by active and effective participation in team discussions; by taking initiative in contributing to the written documentation for the project; and in preparing for and participating in oral presentations.

Keeping a Daily Journal or Research Notebook

You are part of a team, working with other team members on a project that is almost surely too complicated for one person to accomplish in the allocated time. So keep a journal! Your journal can be in a notebook or three-ring binder (your advisor or the Clinic director may mandate a particular format). It need not be elaborate—you could just enter what you have been reading and note difficulties encountered and what you have accomplished towards completion of your part of the project.

The journal or notebook will help you remember what you have done, and will mark the milestones as you progress through your assignments. Your journal or notebook will also be helpful to your team leader and your faculty advisor, who may ask to review your journal or notebook periodically to see how your work is fitting in with the rest of the team and whether additional resources or even redirection of effort is called for. Appendix E has more detail about how and why to maintain a research journal or notebook.

An outstanding student will keep an up-to-date journal or research notebook with sufficient detail to ease the process of project documentation. You should keep notes that make it easy for your team leader and faculty advisor to follow your progress and take appropriate actions when unforeseen problems arise.

Taking Responsibility

In a Clinic project you are expected to demonstrate initiative beyond what is normal for a lecture or laboratory course. Your advisor is available to serve as a resource for the work and help you and your team set priorities, but the responsibility for keeping up the pace of your work, solving day-to-day problems and ensuring that your assignments in the project are completed on schedule lies with you. You are the one who must anticipate and compensate for the added effort you will need in helping prepare the final written report and oral presentation. You own the problem—it's your responsibility!

A Regular Academic-Year Clinic Project Is a Six-Unit Course

A Clinic project is a year-long, regular, six-unit academic course. Treat it as such! If you fulfill all of the criteria listed in this chapter well, but only participate for a fraction of the agreed-upon time, you will not be perceived as a solid performer and will be graded appropriately.

10.2 Final Thoughts

The most important attitude you should maintain is looking forward with pride to a job well done!

Clinic projects require hard work and—most of the time—can be very enjoyable. But there is no guarantee of success in any research program. Failures can and do occur.

At times the stress and disappointment when research doesn't go as planned can be difficult to endure, but things generally have a way of working out in the end. Be assured that the sponsors are usually very pleased by the results of Clinic projects—even negative results because such results can save the sponsor millions of dollars by mapping out fruitless avenues for exploration. If you work conscientiously, you will both learn a great deal along the way and enjoy the satisfaction of completing an important job.

Appendix A

The hmcclinic Document Class

LaTeX uses *document-class files* to define many fundamental aspects of a document, including font sizes; typefaces; predefined commands and environments; the formatting of various elements (e.g., title pages, lists); and other customizations.

The use of a document-class file allows the writer to concentrate on the *content* of their document, leaving the formatting to the document class. It also provides a layer of abstraction, as in most cases it is possible to change only the document class and obtain a typeset document that appears very different from the same document typeset with another document class.

To help you format your Statement of Work, midyear, and final reports, the department has created the hmcclinic document class, which provides a title page (for the midyear and final reports), font selection, and loads a set of core LaTeX packages to provide support for other aspects of the writing process.

A.1 Getting and Using the hmcclinic Document Class

The hmcclinic class can be used on any mathematics department Linux system without any configuration. For other systems, you can download the class file, some basic templates, and some supporting files from https://www.math.hmc.edu/computing/support/tex/classes/hmcclinic/. These files are also available from any mathematics department Linux system or Mac OS X desktop in `/shared/local/texlive/texmf-local/tex/latex/hmcclinic/`. (Chapter 7 has more information about installing and using LaTeX class and package files for writing reports.)

Those locations also contain the file `sample-clinic-statement-of-work.tex`, which gives you an outline of the commands supported by the class file and provides you with a typesettable example.

A.2 Options for the `\documentclass` Command

Document classes typically define *document-class options* to turn some functionality on or off.

The hmcclinic class provides several document-class options. Some of these are mandatory—typesetting your document without them will trigger an error message and stop compilation—and some of them are optional, but help format your document correctly for specific purposes.

A.2.1 Departmental Document-Class Options (Mandatory)

Because the hmcclinic class is used by multiple departments, you must specify a document-class option to set your department in the `\documentclass` command. For math, you'll put

```
\documentclass[mathematics]{hmcclinic}
```

The available department document-class options are

- `biology`
- `computer-science`
- `cs`
- `chemistry`
- `engineering`
- `math`
- `mathematics`
- `physics`

A.2.2 Report Format Document-Class Options (Optional)

In addition, hmcclinic provides two document-class options to help your format your reports for specific uses.

A.2.2.1 The `proposal` Option

For your Statement of Work, specifying the `proposal` document-class option switches the class between using the book and report class for its basic format. The result is a document whose format is more suited for the length of a Statement of Work.

Use it as

```
\documentclass[proposal,<<department>>]{hmcclinic}
```

A.2.2.2 Midyear Report Options

Depending on your department's practice, your midyear reports might be formatted like the final report—like a book—or be a shorter document more akin to the Statement of Work.

The hmcclinic class provides document-class options for specifying which sort of report you're writing. Check with your advisor and Clinic director to find out which option you should use.

All midyear report formats generate a title page similar to the title page on your Statement of Work and final report.

midyearreport The midyearreport document-class option tells the class to format your report the same way that your final report will be formatted. The top-level sectioning command will be \chapter, and you will get all the extra bells and whistles, such as the abstract and acknowledgments environments.

Your report will also be labeled as a "Midyear Report" below the logo on the title page (where the sponsor's name appears).

midyearupdate The midyearupdate document-class option tells the class to format your report as a shorter document, like an article. The top-level sectioning command will be \section rather than \chapter, and the contents of the abstract and acknowledgments environments will be thrown away.

Your report will be labeled as a "Midyear Update" below the logo on the title page (where the sponsor's name appears).

midyear (Deprecated) Prior to 2015, the midyear option did what the midyearreport option now does. Just midyear will work for the time being, giving you the same behavior as midyearreport, but this option may be removed in the future—use midyearreport or midyearupdate instead.

A.2.2.3 The amsthm Option

Most Clinic reports do not include formal statements and proofs, so there's no need for the amsthm package (which provides support for formatting

theorems and theorem-like environments, as well as proofs).

If you do need amsthm for some reason, you should use the `amsthm` document-class option to tell the class to load amsthm for you rather than load it yourself in your master document with `\usepackage`.

Having the class file load the amsthm package avoids errors caused by amsthm trying to redefine commands created by the newpxmath package, which provides the math fonts for the hmcclinic class.

A.3 Cover-Page Commands (Mandatory)

The hmcclinic document class supplies a number of commands for specifying information about your project and your team. As these commands are used in setting the cover page for your report, these commands *must* be specified.

A.3.1 `\sponsor` (Mandatory)

The name of your sponsoring organization (company, research organization, laboratory, college, etc.). Used as

```
\sponsor{Sample Examples, Inc.}
```

A.3.2 `\title` (Mandatory)

A title for your report. Probably something that captures the essence of your project. Used as

```
\title{An Exploration of Complex Mathematical
  Phenomena in the Sample Industry}
```

A.3.3 `\author` (Mandatory)

The names of the members of your team *in alphabetical order*. Names are separated using the `\and` command. The name of the Project Manager should be followed by the string "(Project Manager)". So, for example,

```
\author{Roger Crenshaw \and
  Ellen Henery (Project Manager) \and
  Samantha Smith \and
  Xavier Zavery}
```

If you have multiple project managers during the course of the project (e.g., one for fall and one for spring), you can indicate their term of service as

```
\author{Roger Crenshaw \and
  Ellen Henery (Project Manager, Spring) \and
  Samantha Smith \and
  Xavier Zavery (Project Manager, Fall)}
```

Students who only participate in a project for part of the year should just be listed as usual (without any reference to when they worked on the project).

A.3.4 `\advisor` (Mandatory)

The name of your team's faculty advisor (should you have more than one, use the `\and` command to separate them). Note that you should only include their name, not their title. So,

```
\advisor{George Washington}
```

A.3.5 `\liaison` (Mandatory)

The name(s) of your liaison(s). As

```
\liaison{Thomas Jefferson \and Alexander Hamilton}
```

As with your advisors, don't include titles or degrees.

A.3.6 `\consultant` (Optional)

Some projects may have additional consultants who don't work for the college or the sponsor. Such individuals can be specified using this command, as in

```
\consultant{Abigail Adams}
```

As with your advisors, don't include titles or degrees.

A.3.7 Exceptions to Standard LaTeX Usage

1. Most LaTeX documents define a `\date` command to specify a date. While the hmcclinic class supports the `\date` command, we generally suggest that you not specify a date, or not specify a date until the final version of your document is frozen, so that the date that the document is typeset on will be used instead.

2. The `\eqref` command provided by the amsmath package ordinarily formats equation references as the equation number surrounded by parentheses—for example, as "(1.3)". The hmcclinic class redefines this command to conform to our standard reference-formatting expectations, so `\eqref` will instead create references that look like "Equation 1.3".

A.4 Default Output Format

Recent releases of TEX systems use the pdfTEX engine as the basic processing application for TEX and LATEX documents. As pdf (Portable Document Format) files are also smaller than PostScript versions of the same document, and pdf documents are also better supported by free or integrated viewer applications across all computing platforms, we encourage you to develop your documents exclusively using the `pdflatex` program for compilation.

Note that, whatever format you may have used for development, the final versions of your documents for sharing with your sponsors *must* be pdfs.

Note that the pdfTEX engine only supports graphic file inclusions in pdf, jpeg, or png formats. If you have eps (Encapsulated PostScript) images, you will have to convert them to pdf before they can be used. The `rubber` tool, a command-line script that should be available on most Linux/Unix systems in the department, can handle the conversion for you.

A.5 Basic Appearance of Documents

As the same questions crop up year after year, we will describe some basic fundamentals about the appearance of documents produced with the hmcclinic class. These are key elements of the class and *should not be adjusted or overridden* by your own LATEX code. If you're curious about the specific choices that have been made, please feel free to discuss them with the department's system administrator.

A.5.1 Fonts

We use the newpxtext package to typeset the document using a typeface based on Palatino and a matching set of mathematical fonts (newpxmath and mathalfa).

- The sans-serif typeface is Source Sans Pro; adjusted to work with the x-height of the New PX fonts.
- Double quotes (“ ”) may look reversed; this appearance is a “feature” of the roman typeface.
- `The "typewriter" or monospace typeface is Source Code Pro.`
- The base font size is 11 points, with leading set to 13.6 points (by the standard bk11.clo file).

A.5.2 Page Layout

Pages are laid out using the standard LaTeX book or article classes. Documents are typeset for printing on both sides of the page. All of the laser printers available for student use within the department are duplexing printers.

Chapters and other major sections begin on right-hand pages; that is, it is possible and correct for there to be a blank page on the left-hand sheet in a two-page spread. The blank pages are completely blank—line numbers and headers do not appear on such pages.

Paragraph breaks are indicated by *indentation* of subsequent paragraphs rather than blank lines. Paragraphs immediately following a section header *are not indented*.

A.6 Packages Loaded by **hmcclinic**

The hmcclinic class loads a number of packages in addition to the standard LaTeX kernel code that's loaded by the LaTeX format and the code loaded by the standard LaTeX article or book class that hmcclinic uses as its base.

The packages loaded are:

amsfonts Support for additional fontsets provided by AMS-LaTeX.

amsmath The base package for the AMS-LaTeX system.

booktabs Support for *Chicago*-style tables.

caption Allows us to define the appearance of captions for figures, tables, and other floats. We also provide formatting information for use with the subfig package, but don't actually load subfig in the class.

fancyhdr Used to format the headers and footers of the document.

fontenc Support the T1 font encoding.

graphics Support for including graphics files in various formats in LaTeX documents. Note that the graphicx package provides some enhancements to the commands provided by the graphics package, including the ability to use key=value arguments in the optional arguments of those commands, allowing you to reduce the amount of typing that you would have to do to insert nested commands. If you prefer the graphicx approach to graphics inclusion, just load graphicx in your document's preamble.

helvet Use Hevetica as the sans-serif typeface.

ifthen Support for `if...then` constructs in LaTeX code. Used in some of the class's definitions.

mathalfa Use the STIX BOONDOX calligraphic typeface.

natbib Use the natbib package for citation commands and bibliography formatting.

newpxtext Use New PX as the base (roman) typeface.

newpxmath Use New PX as the math typeface.

textcomp Provides encoding subsets to supply missing characters using the TS1 font encoding.

url Load the url package with the `obeyspaces` and `spaces` document-class options.

verbatim This package provides an enhanced version of the standard LaTeX `verbatim` environments.

xspace Provides the `\xspace` command, which provides a "smart space" that is used at the end of many of our defined commands. (By ending a command definition with `\xspace`, the command can be used as `\foo` rather than `\foo{}` to avoid having the following space eaten by LaTeX.

Appendix B

Installing the hmcclinic Class On Your Own Machine

The mathematics department's computers have the hmcclinic class and other support files installed so that they work without your having to do anything special. You can copy the template file to a new directory and start working immediately—the college's logo and seal, the bibliography style, and some of the configuration options "just work".

If you want to work on your own machine, however, you'll need to have local copies of several different files available for use by your machine's TEX system.

Although it's possible to use LATEX packages or classes without installing them (by placing them in your working directory), we encourage you to install the files into a "TEXMF tree" so that they will be usable by any document you create on your system. Installing support files also allows you to update them in one place, which means it's easy to bring all your documents using a particular class or package up to date, and to be sure that each document is using the latest version of these files that you have installed.

B.1 TEXMF Trees

Modern TEX systems contain thousands of files, including classes and packages, but also several different kinds of font and font-related files; support and format files for bibliographies, indexes, and other document elements; and support for different macro systems built on TEX, including

TEX System	Path
TEX Live (Linux/Unix)	`/usr/local/share/texmf`
MacTeX (Mac OS X)	`/usr/local/texlive/texmf-local`
MikTEX (Windows)	`C:\LOCALTEXMF`

Table B.1 Local TEXMF tree locations for some common TEX systems.

Plain TEX, LATEX, ConTEXt, and others. On the machine being used to write this appendix, there are roughly 66,000 files in the TEX-Live installation.

Each program in a TEX system uses a different combination of these files, and if each program had to search through all the files to find the files it needed, the system would be quite slow. To make the task easier, TEX-system developers created the TEX Directory Structure, or TDS, which defines a standard hierarchical file system to be used to sort components into standard locations.

Furthermore, most TEX systems support multiple TDS-compliant trees that are searched in a particular order, which means that the TEX system can have its own set of files that can be managed separately from those installed by a system administrator or by individuals.

The TEX Directory System is defined and described in a document maintained by the TUG Working Group on a TEX Directory Structure (TWG-TDS, 2004). The whole tree is fairly complex, but most people only need to worry about implementing a small subset of the whole tree.

B.1.1 Local TEXMF Trees

Local TEXMF trees are usually shared by all users of a computer. Their exact location depends on the TEX system in use, but some common locations are shown in Table B.1.

On departmental Mac OS X and Linux systems, the local TEXMF tree is at `/shared/local/texlive/texmf-local`.

B.1.2 Personal TEXMF Trees

Personal TEXMF trees are supported on some systems by default, and on others with special configurations. They're most useful when working on a system shared by other users (such as the math workstations), especially when end users don't have access to system directories, or might want to

TEX System	Path
TEX Live (Linux/Unix)	`~/texmf`
MacTeX (Mac OS X)	`~/Library/texmf`

Table B.2 Personal TEXMF tree locations for some common TEX systems.

override the choices made by system administrators.

For Unix-like systems such as Linux and Mac OS X, personal TEXMF trees can be set up in the user's home directory and are automatically searched. The locations of these trees are shown in Table B.2.

B.1.3 MikTEX Root Directories

The MikTEX system for Windows doesn't support local or personal TEXMF trees by default, although you can set up such directories by using its configuration tools. The Settings tool (in the MikTEX program group) includes a "Roots" tab. You can create a directory (such as `C:\LOCALTEXMF`) and then use the Roots tab to add that directory. The directory you create will have to have subdirectories conforming to the TEX Directory System standard.

Once you've added your new root, you should use the Up and Down buttons to ensure that your root is searched before the other roots. (Alternatively, you can put your root last, and you will always get the version of packages that come with or were installed by MikTEX, but packages you've installed that aren't included in MikTEX's system will be found.

If you add files to your local root after you've added the root to MikTEX's list of root directories, you need to refresh the filename database by hand so that MikTEX's component applications can find the new files.[1]

To update the database, start the MikTEX Settings application and select the General tab. You should see a pane labeled Maintenance, and inside that pane a button labeled Refresh FNDB. Click the button, and wait while MikTEX rescans its file roots. Once it's done, you should be able to use your new files.

Note that you shouldn't need to refresh the database by hand after installing LATEX packages with the MikTEX package manager; the database should be rebuilt as part of the installation process.

[1] MikTEX's filename database is equivalent to the `ls-R` files that Unix TEX systems use; on Unix systems, local and personal TEXMF trees are searched, so you don't need to index them.

B.2 Obtaining the Class and Support Files

There are two basic elements that you will need to produce your final Clinic report:

1. The class and its support files, which include the HMC seal and logo and some configuration files; and,
2. The `hmcmath.bst` bibliography style file.[2]

You can either download a `tar` file or ZIP archive containing the necessary files or copy them from a mathematics department computer.

B.2.1 Downloading the Class and Support Files

The hmcclinic class is available from https://www.math.hmc.edu/computing/support/tex/classes/hmcclinic/.

The bibliography style files can be downloaded from https://www.math.hmc.edu/computing/support/tex/bibliography-styles/.

There are also archives that contain the class and its support files and the bibliographic style files already arranged into the proper directories. If you don't already have your own TEXMF tree, we recommend that you get one of these archives rather than the separate pieces. The `hmcclinic-texmf` archives are available from https://www.math.hmc.edu/computing/support/tex/classes/hmcclinic/.

B.2.2 Copying the Class and Support Files

The class, bibliography-style, and support files can also be copied from any departmental Linux system. The latest version of the class file and support files can always be found in `/shared/local/texlive/texmf-local/tex/latex/hmcclinic/`. The bibliography-style files can be found in `/shared/local/texlive/texmf-local/bibtex/bst/hmcmath/`.

B.3 Installing the Class and Support Files

As noted in Section B.1, there are many possible subdirectories for local and personal TEXMF trees. For installing the basic support files for use with

[2]There is also a `hmcmathannote.bst` file that provides an annotated version of the hmcmath bibliography style, which might be useful if your advisor wants an annotated bibliography; see Section 7.7.3 for details.

```
texmf/
   bibtex/
      bst/
         hmcmath/
            ...
   tex/
      latex/
         hmcclinic/
            ...
         hmcthesis/
            ...
```

Figure B.1 Thesis and Clinic-related TEXMF subdirectories.

the hmcclinic class you only need a small number of these subdirectories, as shown in Figure B.1.

The `hmcclinic-texmf` archives provide these directories for you, allowing you to copy the `texmf` directory they contain into the correct location on your system. (The archive contains more detailed installation instructions.)

Alternatively, you can create the directory tree shown in Figure B.1 in your local (see Section B.1.1) or personal (see Section B.1.2) TEXMF tree, copy the contents of the `hmcclinic` archive to *`TEXMF`*`/tex/latex/hmcclinic` and `hmcmath.bst` to *`TEXMF`*`/bibtex/bst/hmcmath`. You can also `cd` to the destination directory and unpack the archives there.

B.3.1 Testing Your Installation

If you want to test your installation, you can try TEXing a document using the class or package. If it works, you're good!

If not, you can use some command-line tools to see whether TEX can find your new files. Open a terminal or command window and `cd` into any directory except the one in which you installed the files (your working directory is a good choice, but the test should work in any directory). Run the command

```
kpsewhich hmcclinic.cls
```

The program should print the full path to your copy of the hmcclinic class file. If it doesn't, check to be sure that you have the same directory tree

shown in Figure B.1 and that all the directory names are spelled exactly as shown.

B.3.2 Debugging Your Paths

Linux or Mac OS X systems include a `kpsepath` command that can show you the full set of paths that your TEX system will search when looking for a file.[3] So you might run

```
kpsepath -n latex tex
```

to show the full path that the `latex` program will search when looking for TEX source files (which includes class and package files). The output will look something like the following (note that we have split the output at the colons for clarity; the actual output runs everything together on one line):

```
.:
/home/students/username/.texliveyear/texmf-config/tex/latex//:
/home/students/username/.texliveyear/texmf-var/tex/latex//:
/home/students/username/texmf/tex/latex//:
!!/shared/local/texlive/year/texmf-config/tex/latex//:
!!/shared/local/texlive/year/texmf-var/tex/latex//:
!!/shared/local/texlive/year/texmf/tex/latex//:
!!/shared/local/texlive/year/../texmf-local/tex/latex//:
!!/shared/local/texlive/year/texmf-dist/tex/latex//:
/home/students/username/.texliveyear/texmf-config/tex/generic//:
/home/students/username/.texliveyear/texmf-var/tex/generic//:
/home/students/username/texmf/tex/generic//:
!!/shared/local/texlive/year/texmf-config/tex/generic//:
!!/shared/local/texlive/year/texmf-var/tex/generic//:
!!/shared/local/texlive/year/texmf/tex/generic//:
!!/shared/local/texlive/year/../texmf-local/tex/generic//:
!!/shared/local/texlive/year/texmf-dist/tex/generic//:
/home/students/username/.texliveyear/texmf-config/tex///:
/home/students/username/.texliveyear/texmf-var/tex///:
/home/students/username/texmf/tex///:
!!/shared/local/texlive/year/texmf-config/tex///:
!!/shared/local/texlive/year/texmf-var/tex///:
!!/shared/local/texlive/year/texmf/tex///:
```

[3] On a MikTEX system, all the search paths are visible in the MikTEX Settings application.

```
!!/shared/local/texlive/year/../texmf-local/tex///:
!!/shared/local/texlive/year/texmf-dist/tex///
```

The output of this command may be a bit confusing. Each colon-separated component is a directory that is searched by the program. The `.` at the top of the list is equivalent to the current directory (which is why you can put package files in your working directory and TEX will find them; it's also why we recommended that you run the `kpsewhich` command from a different directory in Section B.3.1). The double exclamation marks (`!!`) at the start of some paths indicate that the program should look for a file called `ls-R` at the root of that tree instead of searching the filesystem for matching files.[4] The doubled slashes at the end of the paths tell the search library to search subdirectories, not just the specified directory.

So in our example, the `latex` program will first look in the current working directory. Then it will look in private directories used by the TEX Live TEX system in your home directory (`~/.texliveyear/texmf-config/tex/latex` and `~/.texliveyear/texmf-var/tex/latex`; where `year` will be the year of release).

Next it will look in your personal TEXMF tree (`~/texmf/tex/latex`). Then it will use the `ls-R` file in `/shared/local/texlive/texmf-local/` to get a list of possible completions in subdirectories of `/shared/local/texlive/texmf-local/tex/latex/` (the `shared/local` tree is available on all math department Mac OS X and Linux systems, and is where the latest version of the hmcclinic class and other support files are kept). After that, the program looks at the main system TEXMF tree, again consulting the `ls-R` file.

After searching the `tex/latex` trees, the program will look for files in the `tex/generic` subtree (files that can be used by LATEX or TEX), and then for files meant for use by TEX alone.

If your local or personal TEXMF tree doesn't appear in the list, you may need to do some additional configuration. Please consult the department's information technology analyst or the `latex-l` list (see Section 7.9) for help.

B.3.3 More Information

If you're really curious about how TEX searches for files, you should read the article "A Directory Structure for TEX Files" (TWG-TDS, 2004) and the

[4] These `ls-R` files are generated by the `texhash` program, and are similar to the output of the `ls` command with the `-R` (recursive) flag. Having the system's files be indexed in this way makes TEX much faster.

"Kpathsea Library" manual (Berry and Weber, 2008).

Appendix C

Commonly Seen Problems

Unfortunately, year after year we see some of the same issues with the LaTeX code submitted by students. To help you not make the same mistakes others have made, this appendix explains some of the most commonly seen problems and how to avoid them.

Some of the issues discussed here are fairly obvious, but some deal with the intricacies of writing LaTeX code and may not be at all obvious.

C.1 Spell Checking

A surprising number of final reports contain spelling errors.

Use a spell checker to check your entire document. There are LaTeX-aware spell-checking programs available on all platforms. Some editors also include spell-checking as you write, which is especially useful. If you need advice on choosing an editor or a spell checker, ask!

C.2 General Style and Code Format

The sample thesis/Clinic report code provides an example of how best to format your LaTeX source code for easy reading and editing. It also includes many examples of complex LaTeX structures and has pointers to relevant references. More information about obtaining the sample thesis/Clinic report is available in Section 7.2.2.

C.2.1 Hacking at the Document Format

A major problem that we see occasionally is an attempt to make the typeset thesis or Clinic report look like a Word document. Examples include adding line-break commands to force space between paragraphs; use of `\noindent` or `\indent` to suppress or alter LaTeX's paragraph indentation; and the addition or suppression of blank pages between chapters.

Don't. If you're not sure whether your document looks right, download one of the typeset sample thesis or Clinic reports from https://www.math.hmc.edu/computing/support/tex/sample-report/. If you're still not sure, put a copy of your LaTeX code in your math account and send e-mail to the department's sysadmin, who will take a look at it and let you know if it's okay.

C.2.2 The LaTeX Reference System

We sometimes see hard-coded references to equations, sections, chapters, figures, tables, and other elements for which LaTeX provides the `\label` and `\ref` mechanisms. Coding by hand works fine so long as you always remember to go back and correct the references after moving things around, adding new elements, and so on. It's much, much simpler to let the computer do the tedious work of tracking these changes, especially as these values can be tricky to keep track of.

C.2.3 Recreating Existing Structural Commands/Environments

Some people construct proofs and other mathematical structures by hand rather than using the handy environments and other tools provided by the AMS classes and packages.

Don't. If you're trying to do something that seems like the sort of thing that other people would be likely to want to do, but you're not sure where to look for their code, ask.

Google is a good source for answers to questions about formatting. You can also consult any of the books that we recommend on LaTeX. George Grätzer's *Math into LaTeX* (2000) and its replacement, *More Math into LaTeX* (2007), include lots of examples with both the typeset version of something and the source code that produced it.

```
... and so we suggest that
\[
E = mc^{2}
\]
requires us to understand...
```

... and so we suggest that

$$E = mc^2$$

requires us to understand...

Figure C.1 A Run-In Equation.

C.2.4 Not Using Whitespace

Whitespace is your friend! There is no advantage to cramming things together when writing LaTeX code—LaTeX formats text according to its own rules, which includes collapsing multiple spaces or blank lines, which allows you to use whitespace in your code to help make things more clear to you or an editor.

Whitespace and indentation will not only help you figure out what's going on, but it will also make it easier for anyone else reading your code (such as your advisor or the department's system administrator (who will be vetting your code prior to your thesis being accepted)).

Get in the habit of using whitespace now. You'll be glad when you have to reuse some of your older LaTeX code, and any collaborators or editors you'll have in the future will also be very grateful.

C.2.5 Whitespace in Wrong Places

Often an equation, a list, or a code example should appear midparagraph, like a series of comma-separated phrases would. In these cases, you should not have blank lines between the introductory phrase and the `\begin` command for your environment or after the `\end` environment command and the beginning of the additional text.

Figures C.1 and C.2 show a run-in equation and a run-in list, respectively. Figure C.3[1] shows a list where the last item is the last thing in the paragraph, and we begin a new paragraph by including a blank line between the end of the list environment and the new paragraph.

[1]The poem is William Butler Yeats's "The Second Coming", first published in 1920. In most cases, extracts from poetry should be set using the `verse` environment.

```
... therefore,
\begin{enumerate}
\item The first supposition;
\item The second supposition;
\end{enumerate}
suggests that we ...
```

... therefore,

1. The first supposition;
2. The second supposition;

suggests that we ...

Figure C.2 A Run-In List Environment.

```
... thus Yeats suggests that
\begin{itemize}
\item Things fall apart
\item The center cannot hold
\item Mere anarchy is loosed
  upon the world
\end{itemize}

We believe that this conclusion
is, however, a bleak
interpretation....
```

... thus Yeats suggests that

- Things fall apart
- The center cannot hold
- Mere anarchy is loosed upon the world

We believe that this conclusion is, however, a bleak interpretation....

Figure C.3 A Concluded List Environment. (With apologies to William Yeats.)

NFSS Font Command	Obsolete Font Command
`\textit{xxx}`	`{\it xxx}`
`\textbf{xxx}`	`{\bf xxx}`
`\textsc{xxx}`	`{\sc xxx}`
`\textsf{xxx}`	`{\sf xxx}`

Table C.1 New-Style vs. Old-Style Font Commands.

C.2.6 Using Old-Style $$ Display Math Delimiters

LaTeX documents use the `\[ \]` delimiters for setting display math environments in place of the older, TeX-style `$$` delimiters.

There are two major reasons to use the LaTeX-style delimiters rather than the TeX-style delimiters. First, LaTeX's spacing is different than TeX's, especially when some of the AMS packages are in use (as they are with our thesis and Clinic classes). Second, it's not possible for LaTeX classes and packages to alter the display math environments created with `$$...$$`, so math set in different environments may not be typeset consistently.

The basic rule here is to always use the LaTeX commands.

C.2.7 Using Old-Style Font Commands

A related problem is the use of older, deprecated font commands rather than the "new" font commands (from the "New Font Selection Scheme" or NFSS, introduced in 1989). Table C.1 provides a quick overview of commonly used font commands.

The NFSS commands are better than the old-style commands because they are designed to operate orthogonally to one another, and can, therefore, be nested and generate the proper results. Thus

```
\textsc{\textsf{\textbf{String}}}
```

should (given the selection of fonts and font packages that include all the necessary font shapes and weights) produce the word "String" set in bold, small-capital, sans-serif type; that is, as

STRING

Getting the same results with the old-style commands is left as an exercise for the reader.

C.2.7.1 Emphasis or Italics?

Ideally, you should use `\emph{xxx}` rather than `\textit{xxx}`, as the `\emph` command does the right thing when nested; that is,

```
\emph{My favorite movie?  Hmm.  Probably
  Wenders' \emph{Until the End of the World}},
  she thought.
```

would typeset *Until the End of the World* using roman glyphs so it would stand out properly from the rest of the italicized type; as in

> *My favorite movie? Hmm. Probably Wenders'* Until the End of the World, she thought.

C.2.8 Typing Punctuation

LaTeX uses several special code sequences to represent some common punctuation characters. It's important to get these right and to be consistent in their use to produce an attractive document.

C.2.8.1 Quotation Marks

"Quotation marks" are produced using the ‘‘ and ’’ ligatures for double-quotes; ‘ and ’ for single quotes.

The quotation mark key, ", produces ", or "something" in the typeface used by our classes. In other typefaces the left or opening double-quote character may look odder and more out of place (because it might be set as "straight quotes" or the right or closing double-quote glyph's design might be differ more strongly from that of the left double-quote).

C.2.8.2 Dashes

There are three types of dashes: single dash or hyphen, typed as -, used to connect words in a adjectival phrase such as "parallel-processing system"; *en dashes*, typed as --, used to connect ranges of numbers, as in "10–20"; and *em dashes*, typed as ---, and used for longer breaks or interjections—such as this one.

Important Note

Note that none of these dashes should have whitespace surrounding them—they should be run into surrounding text.

C.2.8.3 Periods and Spacing

LaTeX's handling of periods in different contexts can cause subtle spacing issues. By default, LaTeX assumes that a period marks the end of a sentence, and adds a bit of extra "glue" to that space that can "stretch" if needed to help the line fill the available space. But LaTeX can't tell when a period is used to end a title or an initial, and will insert a sentence space instead of a regular word space. Depending on its context, "Dr. John A. Smith" might appear with a wider space than you might expect or might break the line between the title, the initial, or the surname.

You can insert a *tie* to prevent LaTeX from using a stretchy space or splitting things across a line break by replacing the space with a tilde (~) character. Instead of `Dr. A. Smith`, you might type `Dr.~John~A. Smith`, which would produce "Dr. John A. Smith".

C.2.9 Hyphenation Issues

TeX's hyphenation engine is very powerful, but as new words are introduced to the language, the hyphenation patterns that it uses must be updated. In most cases, you won't want to wait until a new release is made, so LaTeX provides some tools to allow you to specify hyphenation patterns for words.

For words that you use a lot, you will want to add a `\hyphenation` command in the preamble of your document. This command takes a space-separated list of words with syllables indicated with dashes; for example,

```
\hyphenation{da-ta-base ho-meo-stat-ic}
```

For words that are breaking improperly (i.e., the word is not hyphenated or not hyphenated properly, and part of it sticks out into the margin (beyond the edge of the text block)), you can try adding "optional hyphenation" commands, `\-`, at the correct locations within the word. If neither approach helps, you may need to rewrite the sentence in order to make it possible to typeset the paragraph correctly.

Note that one way to see "overfull" lines easily is to add the `draft` document-class option to your `\documentclass` command. In draft mode,

LaTeX will put black "slugs" over the text that extends into the margins, making it easier to see.

Fixing hyphenation issues is a fine-tuning step, usually left until very near the end of your writing process, when you worry about where tables or figures fall, how pages break, and so on.

C.2.10 Formatting Tables

Most of the "tables" provided as examples in books and other introductory material on LaTeX are really "tableaus". They use extra vertical and horizontal lines that create distracting visual noise and don't provide any information.

Most published books typeset tables in a sparser, cleaner style, using whitespace to separate columns and lines. Excellent examples and guidelines are presented by *The Chicago Manual of Style* (University of Chicago Press, 2003). This style is supported by the booktabs package for LaTeX, which is loaded by default in our hmcclinic and hmcthesis document classes.

Section 7.5.2.8 talks briefly about the use of the booktabs package for formatting tables. Examples of how to use the booktabs package's commands are included in our sample thesis and Clinic report (see Section 7.2.2) as well as in other LaTeX materials produced by the department.

C.3 BibTeX Problems

Section 7.7 covers our recommended BibTeX usage.

There are three basic categories of issues that we see with the use of BibTeX in theses.

The first, of course, is not using BibTeX at all. The second is issues with formatting the content of BibTeX databases. The third is not fixing problems flagged by the `bibtex` program.

C.3.1 Formatting BibTeX Databases

Section 7.7.4 discusses some recommended ways to format bibliographic databases. The key idea is to record as much information as possible in your database. Ideally, you want to have the most complete records that you can possibly put together, with full names for all authors, and complete, properly punctuated, and capitalized titles.

Bibliographic styles vary by publication, with some expecting full names of authors and editors and some using initials. Having full names supports

either style, but only having initials means that you won't be able to fulfil the requirements of publications that demand full names.

Similarly, some bibliographic styles use whatever capitalization you supply for titles, whereas others might downcase all but the first word. One problem with sloppy titles appears when your bibliography records are all sentence-capped but the house style calls for headline-capping. Another is when proper names are used but aren't protected using braces. For example, `{F}ourier`, not `Fourier` (or, worst, `fourier`).

C.3.2 Ignoring BibTeX Errors and Warnings

When BibTeX is run on an auxiliary file, it pulls records out of the associated bibliography database file and formats them according to the specified bibliography file. Along the way, it will generate warnings about missing fields or misformatted information, and errors when important information is missing or duplicated.

Many submitted theses or Clinic reports included bibliography entries with missing dates, journal titles, or other important information. Some contained more than one entry with the same key. And there were other, similar problems.

BibTeX is pretty good about telling you exactly what's wrong, including the line number and the record that has the problem, so there's really no excuse for not fixing these issues.

Appendix D

Creating CD-ROMs and Your Electronic Archive

Two of your major deliverables are

- A CD-ROM containing the final version of code, the final version of your various reports, as well as white papers and other materials you may have generated during your project that would be of use to your sponsor.
- An electronic archive containing the same information, plus other materials you're not sending to your sponsor but that will be saved for Clinic records.

We recommend that you create both sets of deliverables at the same time by populating a `/home/clinic/`*`year`*`/`*`project`*`/cdrom` directory with the various pieces for the CD-ROM, then generating a CD-ROM image from that directory and burning CD-ROMs using that image.

Burning the CD-ROMs should be the last thing that you do, as they will need to include a PDF of the very final version of your final report. But you can get started with the archive at any point (even during the first week of your project!).

D.1 CD-ROM Contents

Details on the contents of the CD-ROM are in Section 9.5. There is a set of required items (in Section 9.5.1) that should go into a `cdrom` directory in your project directory, something like

/home/clinic/*year*/*project* your Clinic directory

cdrom directory for assembling your CD-ROM

sow copy of your Statement of Work in PDF

reports midyear and final reports in PDF

code any computer program source code you wrote (C++, Java, MATLAB, FORTRAN, etc.), with subdirectories for each project if you had more than one—see Section D.4.

presentation your final presentation slides

poster your final as-printed poster

You can then use this directory to create your CD-ROMs for your bound reports.

You may also include optional materials on your CD-ROM, which might include any of the things mentioned in Section 9.5.2. Of particular interest to your sponsor would be white papers or other materials that you generated while working your way through the problem that aren't already incorporated into your final report.

Important Note

Unlike your Github repository, which should *not* contain compiled PDF versions of your reports, your CD-ROM *should* include both the LaTeX source code and the compiled PDF versions. see the discussion in Section D.4 for details on how to get the code.)

D.2 Departmental Archive Contents

In addition to the `cdrom` directory with the materials you're sending to your sponsor, the department expects you to include all the supporting materials you generated in the `/home/clinic/`*`year`*`/`*`project`* archive, so be sure to make that material available.

Important Note

If those materials are in your wiki or Github repo, you don't need to include separate copies of them for the electronic archive; we'll either maintain the wiki site and repositories or dump them later.

In other words, you don't need to include code that you're not putting in your `cdrom` directory in the rest of your archive.

D.3 Data Sets or Other Large Sets of Files

If your project generated large amounts of data or other files that your sponsor will need that won't fit on a CD-ROM, let us know and we can explore some other alternatives. If they can regenerate the data, it doesn't need to be included.

Note that some datasets may have been obtained under licensing terms that prevent the material from being given to or shared with the sponsor (e.g., commercial data products on CD-ROM or DVD-ROM).

Important Note

Do not include licensed information in your CD-ROM, copy it to your `/home/clinic` directory, or send any media to your sponsors, but be sure to turn in any media to the Clinic Coordinator or let us know so that we can make backups and store them safely.

D.4 Exporting Code

Where you have code that you've been working on, you should include a copy of *the final version of that code*. (This includes the LaTeX source code for the final versions of your reports.)

Git allows you to "export" code from your repository so you have a clean copy of the code without any of the metadata that Git uses to keep track of changes (i.e., without the `.git` directory).

To export code, you `cd` into your working copy and then run `git` as follows:

```
git checkout-index -a -f --prefix=/path_to_destination/
```

where *`path to destination`* is the full path to the location you want your exported code to end up. For example,

```
git checkout-index -a -f --prefix=/home/clinic/year/team/repo_name/
```

would put the latest version of your code in a directory called *repo name* inside your Clinic home directory (Section 3.6.1).

Important Note

Notice the ending slash (/) after *path to destination*. It's very important to ensure that you have that slash, otherwise the files will end up in the directory above your desired directory, with the last part of the path prepended to the name of each file!

For example, assuming a working copy called `work` with three files, `foo`, `bar`, and `baz`, running

```
git checkout-index -a -f --prefix=/tmp/work/
```

will create the directory `work` in `/tmp`, and inside that directory you'll find `foo`, `bar`, and `baz`.

But running

```
git checkout-index -a -f --prefix=/tmp/work
```

will get you

```
workfoo
workbar
workbaz
```

in `/tmp`.

D.5 Changelogs

You might want to include a changelog file for each export project, so that people can see the commit logs for your work.

For each repo, you would `cd` into your working copy and run

```
git log > changelog
```

You will, of course, have to copy that file into your exported repo.

D.6 Creating CD-ROMs

Section 9.5 mentions that you should ensure that your CD-ROMs are readable on Macs, Linux machines, and Windows machines. Recent Windows machines use a format that isn't widely supported by other operating systems, and creating CD-ROMs on Linux isn't the easiest thing in the world, so I would encourage you to use the Macs to create your CD-ROMs.

Because burning the CD-ROMs requires a certain amount of calculation each time, we recommend that you first create a disk image file, verify its contents, and then use that image to create your physical CD-ROMs.

D.6.1 Creating a Disk Image

Once you've finished assembling the contents of your `cdrom` directory,

1. Copy it to `/tmp`[1] on a Mac with, for example,

   ```
   cp -r /home/clinic/year/project/cdrom
   /tmp/project-year-cdrom
   ```

   (All on one line; the break is to make it easier to read.)

   Note that the name of this directory will be used for the name of the disc that you create, so giving it a meaningful name is important.

2. Ensure that the permissions on the contents are correct by running

   ```
   chmod -R go+rwX /tmp/project-year-cdrom
   ```

   Verify that the permissions are right by running "ls -l" on the directory and maybe on some random files; you should see something like

   ```
   drwxrwxrwx 10 you group 340 May 1 20:30 project-2011-cdrom
   ```

   with similar permissions on directories within that folder. Files should look something like

   ```
   -rw-rw-rw-  1 you  group  1568941 Apr 29 17:50 final.pdf
   ```

   You could also have someone else log into the machine and check the files (e.g., read text files in the Terminal or with a text editor; open PDFs with Preview); if they can read them, you're good.

[1] If you reboot the machine, anything in `/tmp` will be removed.

3. Run Disk Utility (in `/Applications/Utilities`).
4. Choose File→New→Disk Image from Folder....
5. Go to `/tmp` by pressing Command-Shift-G, and typing `/tmp` in the "Go to the folder:" sheet that drops down).
6. Select the directory you created in step 1.
7. Click the Image button.
8. Select a directory on the local machine (e.g., `/tmp`) to save the image.
9. Set the name to something sensible, such as *`project-year-`*`cdrom.dmg`. (The file name is just for you.)
10. Set Image Format to "hybrid image (HFS+/ISO/UDF)".[2]
11. Click the Save button.

You will end up with `.dmg` file containing the different formats. You can double-click on the image file in the Finder to automatically mount it and verify that everything you want on your disc is in the image.

D.6.2 Burning CD-ROMs

Then, to burn CD-ROMs,

1. Select the disk image from the list in the left pane of Disk Utility (or drag the image into Disk Utility if it isn't there)
2. Click the Burn button in the toolbar, which should prompt you to insert a blank CD-ROM

Now would be a good time to verify that your burned CD-ROM is readable on Linux and Windows machines before you make additional copies. In particular, it should mount automatically (without having root privileges) on modern Linux machines and the contents should be readable.

When you're ready to burn the remaining disks for each copy of your report, go back to Disk Utility and continue to burn CD-ROMs with the Burn button.

[2]Mac OS X uses HFS+; most Unix systems use ISO format; Windows prefers the newer UDF format, which may also be readable on other systems.

Appendix E

Guidelines for Maintaining a Technical Project Notebook

A project notebook is

> the medium by which a researcher organizes and focuses his thoughts, keeps records of daily work, and provides a record of tasks performed. Notebooks record investigative history, both successes and failures, so that research does not have to be repeated. These notebooks represent the factual and legal documentation of work performed by technical personnel within the organization.
>
> —Dawn Duperault, School of Library and Information Studies, Texas Woman's University, 1995

E.1 Why Keep a Project Notebook?

While the ubiquitousness of laptop computers, Wikis, note-keeping software, internal blogs, and other electronic record-keeping technologies has caused the whole culture of handwritten project notebooks to evolve, these physical, paper-and-ink documents still play a vital role in the research process. Notably, the paper-based project notebook is still essential in the patent process and provides a time-honored (and legally admissible) way of documenting scientific discovery.

Early in the semester (by Clinic/Sponsor Orientation Day, September 6 (Thu), at the latest), you and your teammates, your advisor, your project

manager, and your sponsor's liaison should meet and agree on the exact nature of how project notes will be kept for your Clinic project. The guidelines for written notes included in this chapter are the traditional rules for such documents, and following these rules or similar variants will be expected by many of our sponsors.

Keeping a project notebook can be justified on at least five levels:

- Technical
- Personal
- Academic
- Professional
- Legal

E.1.1 Technical Reasons

A project notebook documents the progress of your experiments (basic research, product, or software development), observations, and results.

A notebook makes the process of archiving data more efficient. It also creates a resource for interpreting your results and planning future experiments.

E.1.2 Personal Reasons

Personal reasons for maintaining a project notebook include

- Developing your skills of observation, ordering evidence, and drawing inferences as well as your descriptive powers
- Helping you "remember" everything that you do for the entire length of the project
- Avoiding duplication of effort
- Avoiding repetition of erroneous procedures
- Insuring against the loss of valuable data

E.1.3 Academic Reasons

Academic reasons for maintaining a project notebook include

- Providing you with a solid record of your participation in a project
- Documenting the contributions of team members on a project
- Providing you with the information you will need to prepare technical reports or journal articles at a later date

E.1.4 Professional Reasons

Professional reasons include

- Supporting the validity of results reported to peers and sponsors
- Providing proof of fulfillment of contracts
- Maintaining continuity when a long-term project is passed from one team to another

E.1.5 Legal Reasons

Legal reasons include

- Establishing a permanent record to prove the authenticity of the work
- Availability as evidence to prove a company's or college's right to obtain a U.S. patent
- Defending a patent against those who contest it
- Defending a patent against infringement by others

E.1.5.1 Inventions and Patent Law

The dates of "conception" and "reduction to practice" are essential to obtaining and defending a patent.

Date of Conception Record the novel idea in the notebook as quickly as possible

Evidence of Due Diligence Continue to record every instance when you return to the idea

Reduction to Practice Established by the actual construction and successful testing of device incorporating the invention

E.2 Notebook Management

In order to serve as evidence, a project notebook must be difficult or impossible to fake. Thus, it should be permanently bound with numbered pages (so that adding pages is difficult or impossible, and missing pages will stand out).

Each team member should have her own notebook, and investigators should only make entries in their own notebooks. Having multiple investigators record the same information is okay—you will have duplicate records should one notebook be damaged or lost.

Legibility is important—the greatest discovery in the world might not be upheld if no one but the investigator can read the description in the project notebook!

All entries should be made in ink. Ideally, you should avoid using different colors of ink on the same page (or even on the same day). Using the same color ink helps make it clear that the entries were made at roughly the same time.

For the same reason, you should avoid leaving blank pages between entries or even significant amounts of blank space on a single page. Blank pages or spaces could be used to modify your records later.

All entries should be made consecutively, in the order in which the events that they describe occurred. Save nonchronological organization methods for your reports or articles.

Reserve the first few pages of your notebook for use as a table of contents—update that section daily with pointers to significant events, when new days' entries start, and so forth.

Don't try to erase or excise errors. Instead, cross them out with a simple line (i.e., don't scribble wildly over the deleted material or black it out with a marker) and write the corrected information in after the deleted section.

If you discover an error in your work on a following day, cross out the erroneous material and add a note that points to the corrected information. Include the page number where the corrected material is written and sign and date the deletion.

Notebook Entries To Date	Contact Information
Meeting Minutes	Project Description
Individual Log	Goals and Objectives
Group Log	Schedule
Name and Location of Computer Files	Short-Term Plan
Correspondence	Long-Term Plan
References	Work Schedule
Team Members	

Table E.1 Notebook Entries to Date.

E.2.1 Signing and Dating Pages

Every page of your notebook should be signed and dated. Table E.1 includes a list of specific types of information that should be dated independent of their position on a page.

E.2.2 Supplemental Material

Materials that are generated by equipment or computer programs that support your work should be permanently inserted into your notebook at the appropriate location. They should be attached using tape or glue along their left-hand margin, and should be signed and dated with the signature extending across the attached material and onto the notebook page.

E.2.3 Computer Records

If your team is writing computer code, you should use a *version-control system* to maintain that code, and perform regular check-ins of your code with meaningful commit messages. Section 4.8 talks more about the use of version-control systems

E.3 What to Record

The project notebook is a daily record of your activities—what you are thinking and doing.

- Ideas, calculations experimental results, and observations should be entered into the notebook on the same date they occur

- Results include more than data. Record questions, reasons for changes in experimental plans, evolving thoughts, awareness of potential of discoveries, and so forth.
- Exactly what was done—protocols, design of experiments, calculations, etc.
- Why it was done (e.g., objectives and goals)
- Who initially suggested the idea
- Who did the investigation/exercise/task/test
- When and where the task was performed
- The results obtained
- The conclusions drawn

At the end of a work day,

- Summarize what you have accomplished
  - Conclusions are not necessary
  - The summary helps maintain continuity, reveals where the work left off and how it might resume
- Periodically have your work witnessed
- Store your notebook in a safe place

E.3.1 Witnesses and Reviewers

Must be adults, preferably over 21, with the technical competence to understand the details of the subject

Must be an impartial person—not a co-inventor

Sign and date every page reviewed using a form similar to

Witnessed and understood by _____, Date _____.

E.4 Things to Avoid

Problems with your notebook may reduce its value in legal or academic disputes or damage the credibility of your laboratory or team.

These problems include

- Illegible entries
- Unsigned or undated pages
- Pages that have not been witnessed
- Long delays between the signing of the page by the inventor and the witness
- Consecutive notebook pages that are not dated in chronological order
- Missing notebook pages
- Erasures and deletions

E.5 Conclusion

Remember, a project notebook

- Says exactly what was done, when it was accomplished, and by whom the work was performed
- Reveals the thought process of the investigator in the conception of ideas and the interpretation of results
- Enables someone else to repeat the work at some future date
- Is durable and verifiable

E.6 Acknowledgments

This chapter is based on slides from a presentation made by Professor Kerry Karukstis to the mathematics Clinic on November 20, 2001. Additional material was written based on BookFactory (2007); Caprette (2005); Garabedian (1997); Purrington (2011).

Appendix F

Project Management: A Job Description

Project management is a challenging and rewarding role. Not everybody can (or wants to) do it. Here are some thoughts about Clinic projects and project management that will help you decide whether you want to throw your hat in the ring. If you are selected as a project manager, here's what will be expected of you.

Clinic projects are too difficult for an individual to accomplish in the allotted time, so teamwork is essential for a successful project. The project manager is the person who is in charge of maintaining coordination of a project. It's a big responsibility. Let's put it in perspective.

Your one-year, six-unit Clinic course will be over almost before you know it. And, if you are a senior like most Clinic participants, the last weeks of the spring semester will be consumed by preparing for graduation and applying for graduate school and/or summer employment. On top of all that, your team will have the difficult job of completing your Clinic project, preparing the final presentation for Projects Day and the Final Report. So it is very important that your team be well-coordinated and working efficiently towards completion of the project, else chaos rules.

One of the first things your project team will do is write a Statement of Work that explains how your team will accomplish the work proposed in your sponsor's project proposal. The Statement of Work is a contract, and you will negotiate its final form with the persons appointed by your sponsor as the project liaison.

The success of a Clinic project depends in large part on several factors, including

- How well the Statement of Work and its objectives are matched to the skills of the individual team members
- How effectively the disparate activities of the team members are coordinated towards achieving the objectives of the work statement
- How effectively the team is guided towards completing work on schedule
- How well the team efforts are communicated to the sponsoring liaison
- How well the team responds to the suggestions of the faculty advisor and liaison

These factors are summed up by the word "organization". Good organization does not often arise spontaneously. Ordinarily, it is brought about by selecting an individual to take responsibility in coordination, scheduling and communications on behalf of the project. That person is the project manager.

Here are some of the things a project manager must do:

- Accept responsibility for the coordination of a team
- Work to promote a sense of shared enterprise and cooperation among the team members
- Understand the distinct skills of each team member and how each one may best fit within the project
- Seek and promote consensus among team members in defining project objectives
- Be fair-minded in apportioning responsibilities to carry out the tasks needed to accomplish the objectives
- Gain acceptance of responsibility of each team member in performing assigned tasks
- Set goals and milestones to assure progress towards scheduled tasks
- Pay meticulous attention to the details of a project

- Take timely action when it becomes clear that milestones are not being met, assess cause and take corrective action
- Negotiate alternative research objectives with sponsor if/when it becomes clear that a prior objective will fail
- Be prepared to make hard decisions when unresolved personnel problems arise
- Maintain continuing good communications with the liaison and faculty advisor
- Keep an eye on the calendar to ensure timely delivery of all promised work products

These tasks are not the kinds of things that everyone can do or will want to do. But, for the kind of person who wants to take on the extra responsibility, the special satisfaction that comes with keeping a project running smoothly and on track is a rare and precious reward that makes the effort worthwhile.

It will look good on your resume, too.

Appendix G

How to Give a Good Talk

By Joseph A. Gallian.
Reprinted from *Math Horizons* magazine (1998).

∞ ∞ ∞

Presentations by undergraduates at professional meetings have increased dramatically in recent years. And more and more undergraduates make presentations in classes, seminars and colloquiums. Learning how to give a good presentation is a valuable skill that many students will find useful in connection with their employment. Baseball manager Joe Torre once said that the teams that are most successful are the ones that do the little things well. Doing the little things well is the secret to giving good talks as well. Here is my advice on how to do the little things well when giving a talk.

Preparation

- Inquire about the target audience.
- Determine the level of knowledge of the target audience.
- Choose a subject that will appeal to the intended audience.
- Don't overestimate what the audience knows about your subject.
- Don't try to do too much.

- Use simple examples and concrete special cases. A "non-example" often helps to clarify a concept. (For instance, if you use the integers modulo 7 as an example of a finite field, be sure to point out that the integers modulo 6 is not a field and why.) Use intuitive definitions rather than technically correct ones. Avoid details. Mention applications.
- Choose a short and informative title. (Cute titles are usually poor titles.) "On a Theorem of Hilbert" is too vague. "On Hilbert's Basis Theorem" is short and informative. In your abstract, indicate the level of the talk. (Examples: "This talk is intended for a general audience"; "This talk is suitable for those who have had linear algebra"; "This talk is suitable for those who have had real analysis.")
- Keep technical terms and unfamiliar symbols to a minimum. When you do use them remind your audience of their meaning from time to time.
- When possible, relate your topic to other fields.
- Provide a context for your talk. Explain how you got interested in the subject. Mention others who have worked on the subject of your talk.
- Use transparencies. A chalkboard talk comes across as slow moving. Make the transparencies up well in advance and number them.
- Use multi-colors in preparing your transparencies. Blue, green, red and purple show up best. Avoid orange and brown. Use permanent ink (water soluble smudges easily). You can fix mistakes with rubbing alcohol.
- Write very large or use a large font (even for a talk in a small room). If you reproduce printed material, enlarge them for the transparencies. Use color photocopy machines to make color transparencies.
- Prepare a crisp beginning. Perhaps start with a question, an application or a prop.
- Don't put much on the transparencies. Use key words and phrases instead of entire sentences. Avoid filling transparencies with equations and formulas. (Don't compute in public.)

Appendix H

Teleconferencing Etiquette

The following are some useful tips for maximizing the usefulness of teleconference calls (and minimizing their length).

H.1 Before the Conference

- At least 24 hours before a conference, determine an agenda and send it to your liaison and all call participants.
- Make sure that your room is reserved and the telephone is set up and working.
- As a team, develop concrete questions to discuss with the liaison.
- Make sure each team member has at least one piece of information to communicate.

H.2 During the Conference

- Arrive *before* the scheduled time of the call and do any last-minute preparations before you make the call.
- Call your liaison at the scheduled time.
- Introduce yourselves at the beginning of each conference so the liaison knows who is present.
- Identify yourself each time you speak.

- When you have had a difficult week, try to describe what you attempted and what results you obtained, rather than simply saying it was a bad week and nothing happened.
- Avoid side conversations and inside jokes.
- Make sure you speak clearly—take turns and direct your voice towards the microphone.
- Avoid casual language such as “like”, “um”, “cool”, “sweet”, “awesome”. Instead, use more formal language, such as “efficient”, “elegant”, “effective”, and so on.
- Avoid extraneous noise—rustling papers, tapping on the table, typing on loud keyboards, opening or closing doors, and so forth.
- Keep friendly chit-chat to a minimum. End the conference as soon as all topics have been addressed and *no later than the scheduled end time*.
- Express appreciation for your liaison’s insights and time.

H.3 After the Conference

- Post the minutes to your wiki.

Appendix I

HMC Policy on Use of Personal, College-Owned, and Rental Vehicles

The following is based on the *Policy for Travel, Entertainment and Other Business Expenses*, dated February, 2008. It may be superseded by the *Vehicle Insurance Policy*; both policies are available from the Business Affairs Forms and Policies page.

I.1 Personal Automobile

The College will reimburse employees for mileage when a personal vehicle is used on official College-related business that is properly authorized, reasonable and appropriately documented. This does not include mileage for an employee's standard commute when traveling between their residence and the College. The College will reimburse employees at the IRS mileage allowance in effect on the date of travel. The current standard mileage rate can be found on the IRS website using the following link: http://www.irs.gov/Tax-Professionals/Standard-Mileage-Rates

In the event a private vehicle is used in lieu of commercial transportation (air, bus, rail), the lesser of the costs between a coach/economy class tickets combined with other related ground transportation and the mileage would be reimbursed. The College will not reimburse vehicle operating, maintenance or repair costs for personal vehicles (the IRS mileage rate is intended to incorporate these types of costs).

Employees should contact the Human Resources Office (extension 79700) prior to a trip to ensure they are on the College's Approved Drivers List.

The employee's personal automobile liability insurance is the primary coverage when using a personal vehicle on College-related business.

I.2 College-Owned Vehicles (Including Electric Carts)

These vehicles are subject to additional rules pertaining to use. Gasoline purchases with receipts are reimbursable, but mileage is not. Employees should ensure they are on the College's Approved Drivers List prior to a operating a College-owned vehicle. Approved Drivers List information is available by calling the Human Resources Office (extension 79700).

I.3 Rental Vehicles

Employees may rent a vehicle when driving is more convenient than airline or rail travel; driving is necessary to transport large or bulky materials; driving is more economical than public transportation modes due to multiple locations to visit in the destination city; or other surface transportation is not practical. *Employees should ensure that they are on the College's Approved Drivers List prior to renting a vehicle. Approved Drivers List information is available by calling the Human Resources Office (extension 79700).*

I.3.1 Rental Vehicle Insurance Coverage

The College's automotive insurance policy provides primary liability coverage (or secondary coverage behind the rental car company in certain states). It is not necessary for employees to purchase additional liability insurance from the rental car company. Comprehensive and collision losses (damage to rental car) are not covered by the College's automotive insurance policy. When the Wells Fargo–WellsOne College purchase card is used to rent a vehicle, it provides primary comprehensive and collision coverage on the rental vehicle. Thus, employees are encouraged to use a WellsOne College purchase card for rental vehicle purposes, when available. Employees not using a WellsOne College purchase card should purchase comprehensive and collision coverage from the rental car company when renting a vehicle.

Employees not on the Authorized Drivers List expose their personal liability insurance to primary insurance coverage status.

I.3.2 Other Insurance Issues Pertaining to Personal and College Owned Vehicles

Additional information can be found at http://www.hmc.edu/bao/forms-and-policies/

I.4 Ground Transportation, Parking Fees, and Toll Charges

Preferred choices for ground transportation are shuttle service, public limousine service, or other forms of public transportation. Travelers should use their best judgment in choosing ground transportation. Taxi fares require a receipt for reimbursement. Tolls and reasonable parking costs are reimbursable with a receipt.

Appendix J

History

Harvey Mudd College's engineering department began the Clinic program in 1963. It was called a "clinic" because it gave students experience in engineering as practiced by professional engineers, just as a medical clinic provides interns with experience in actual medical practice. At the time, providing undergraduate students with the opportunity to participate in real-world research was extremely rare; such opportunities were usually reserved for graduate school or for a new graduate's first on-the-job experience.

By the fall of 1972, the Engineering Clinic had been in operation for eight years and had proved itself to be an efficient and well-regarded means of involving undergraduate students in open-ended projects that originated in industrial settings.

As funding was obtainable for more projects than engineering faculty were available to supervise, faculty from other departments began assisting with the supervision of some Clinic projects.

During the same period (from 1970–1972), funds from the Sloan Foundation had been used to support Professor Stavros Busenberg and visiting professor William Hall in teaching an experimental seminar in applied mathematics. This seminar centered around the participants working on problems of current interest in industry, similar to the Clinic program's approach. The success of the seminar encouraged the applied-mathematics faculty—especially Professors Busenberg and Robert Borrelli—to decide that it was both desirable and feasible for Clinic projects that were selected and supervised by mathematicians to become a feature of the mathematics curriculum.

The chair of the mathematics department, John Greever, with the support

of the Dean of Faculty, Jacob Frankel, established the Mathematics Clinic despite some initial opposition both within and outside the department. Over a period of months, various details of structure, operation, and budget were negotiated, and administrative approval was ultimately obtained.

J.1 1973: The First Mathematics Clinic

The first Mathematics Clinic project was scheduled for the fall semester of 1973. Professor Busenberg, who was then in his last year as an untenured assistant professor, volunteered to supervise this first project as an overload to his assigned teaching schedule. The project was funded by Bell and Howell Research Laboratories in Pasadena.

Work on the project was carried out by a team of three HMC students during the fall semester, and two in the spring, and resulted in a well-received mathematical model of scintillation phenomena in rear-projection screens.

From the outset, it was clear that while the Engineering Clinic provided an excellent model on which to build, the more abstruse nature of mathematics projects would typically require greater involvement and guidance from the faculty advisor(s).

J.2 Collaboration with the Claremont Graduate School

The Claremont Graduate School (CGS; now Claremont Graduate University), whose mathematics faculty were mostly in pure areas of mathematics, began offering a Ph.D. in mathematics in 1969. By 1972 there was substantial concern about diminishing employment prospects for young mathematicians in non-applied areas, and a concommitant change in funding for graduate programs in such areas.

The CGS mathematics department met this challenge by shifting its emphasis to a master's program in applied mathematics. As a means of supporting this new program and assisting the CGS mathematicians with their financial problems, HMC offered to enlarge the Mathematics Clinic into a joint operation. This new joint Clinic structure was in place for the academic year 1974–1975 and Professor Greever of HMC and Professor Jerome Spanier of CGS were its co-directors.

To avoid confusing prospective clients (who were primarily approached by the HMC Development Office), the façade of a single joint CGS/HMC

Mathematics Clinic was created, although each institution actually maintained its own operation, administrative procedures, and budget. As a further means of supporting the graduate program, HMC began the practice of awarding stipends to a few CGS students who would then act as team leaders for HMC projects.

In the fall of 1974, our second Clinic project was sponsored by Northrop Corporation, supervised by Professor Borrelli of HMC, and carried out by a team of HMC students and one CGS student.

During the spring semester of 1975, we had *two* Clinic projects and CGS completed its first Clinic project with Professor Spanier supervising. The joint Clinic operation continued until 1980, when the HMC and CGS Clinics became entirely independent.

J.3 A History of Success

Since the founding of the Mathematics Clinic program, there have been between two and five Clinic projects every semester. Over a hundred projects have been completed, supervised by at least thirty different faculty members.

We have worked with many different sponsors, many of whom have returned for additional projects or for refinements of previous projects. Projects have been conducted jointly with the physics and engineering departments and have drawn students from every major on campus including our own joint majors in mathematics and computer science and mathematical biology. The mathematics department has always enjoyed a strong relationship with Mudd's computer-science department and our Clinic programs have often been managed in tandem.

Sponsors have included entrepreneurs and national labs, defense contractors, and nonprofit corporations. The projects have studied aspects of astrodynamics, nanotechnology, mathematical biology, data mining, mathematical finance, artificial intelligence, cryptography, criminology, and the mathematics of competitive sports. In short, our Clinic projects have spanned the breadth of mathematics and its applications.

Mathematics Clinic program alumni have gone on to start businesses, make (and lose) millions of dollars, and, perhaps most importantly, to play a role in the sponsorship of many of today's projects. We are grateful for their support and together we will ensure that the Mathematics Clinic will thrive for many years to come.

J.4 Directors of the Mathematics Clinic

The Directors of the Harvey Mudd College Mathematics Clinic to date are as follows:

1973–1974 John Greever

1974–1975 John Greever (HMC) and Jerome Spanier (CGS)—Joint Clinic

1975–1977 Robert L. Borrelli (HMC) and Jerome Spanier (CGS)—Joint Clinic

1977–1978 Courtney S. Coleman (HMC) and Jerome Spanier (CGS)—Joint Clinic

1978–1980 Robert L. Borrelli (HMC) and Jerome Spanier (CGS)—Joint Clinic

1980–1981 Stavros N. Busenberg

1981–1984 Robert L. Borrelli

1984–1985 Stavros N. Busenberg

1985–1990 Robert L. Borrelli

1990–1992 Stavros N. Busenberg

1992–1993 Henry A. Krieger

1993–1999 Robert L. Borrelli

1999–2006 Michael R. Raugh

2006–2007 Weiqing Gu

2007–2008 Andrew J. Bernoff

2008–2009 Alfonso Castro

2009–2010 Weiqing Gu; Susan Martonosi

2010–2014 Susan Martonosi

2015– Weiqing Gu

Bibliography

American Mathematical Society. 1999. *User's Guide for the amsmath Package*. American Mathematical Society.

Berlin, Daniel, and Garrett Rooney. 2006. *Practical Subversion*. Apress, 2nd ed.

Berry, Karl, and Olaf Weber. 2008. *Kpathsea Library*. Included with TEX-Live installation. For version 3.5.7.

BookFactory. 2007. BookFactory laboratory notebook guidelines. Website. URL http://www.bookfactory.com/special_info/lab_notebook_guidelines.html. Viewed on 2007 August 30.

Bringhurst, Robert. 1996. *The Elements of Typographic Style*. Vancouver, BC: Hartley & Marks, 2nd ed.

Caprette, David R. 2005. Guidelines for keeping a laboratory record. Website. URL http://www.ruf.rice.edu/~bioslabs/tools/notebook/notebook.html. Viewed on 2006 June 14.

Carlisle, David P. 1999. *Packages in the 'graphics' Bundle*.

Chacon, Scott, and Ben Straub. 2015. *Pro Git*. 819, Apress. URL https://progit.org/. Home page for Pro Git, which includes links to download the book in various formats as well as links to purchase a physical copy. Page numbers taken from the PDF edition.

Collins-Sussman, Ben, Brian W. Fitzpatrick, and C. Michael Pilato. 2004. *Version Control with Subversion*. Sebastopol, CA: O'Reilly & Associates. URL http://www.svnbook.org/.

The "Subversion Book".

Connelly, Claire. 2006. TEX and LATEX resources. URL https://www.math.hmc.edu/computing/support/tex/. Accessed on 2006 May 2.

Doumont, Jean-Luc. 2009. *Trees, Maps, and Theorems: Effective Communication for Rational Minds*. Kraainem, Belgium: Principiæ.

Dupré, Lyn. 1998. *BUGS in Writing*. Reading, MA: Addison-Wesley, 2nd ed.

Editor. 1986. Hyphenation exception log. *TUGboat* 7(3):145.

Fear, Simon. 1997. *Publication Quality Tables in LATEX*.

Fenn, Jürgen. 2006. Managing citations and your bibliography with BIBTEX. *The PracTEX Journal* 1(4). URL http://www.tug.org/pracjourn/2006-4/fenn/fenn.pdf. Published online.

Gallian, Joseph. 1998. How to give a good talk. *Math Horizons* .

Garabedian, Todd E. 1997. Laboratory record keeping. *Nature Biotechnology* 15:799–800. URL http://www.wiggin.com/pubs/articles_template.asp?ID=102187242000. Republished at Wiggin and Dana website. Viewed on 2006 June 14.

Grätzer, George. 2000. *Math into LATEX*. Boston: Birkhäuser, 3rd ed.

———. 2007. *More Math into LATEX*. New York: Springer Science+Business Media, 4th ed. URL http://www.springerlink.com/content/978-0-387-32289-6/.

Higham, Nicholas J. 1998. *Handbook of Writing for the Mathematical Sciences*. SIAM, 2nd ed.

Jaffe, Greg. 2000. What's your point, Lieutenant? please, just cut to the pie charts: The Pentagon declares war on electronic slide shows that make briefings a pain. *The Wall Street Journal* URL http://interactive.wsj.com/archive/retrieve.cgi?id=SB956703757412556977.djm. From WSJ Online; originally appeared April 26, 2000.

Knuth, Donald E. 1993. *The TEXbook*. Addison-Wesley.

Knuth, Donald E., Tracy Larrabee, and Paul M. Roberts. 1989. Mathematical writing. *Mathematics Association of America Notes* 14.

Kopka, Helmut, and Patrick W. Daly. 2004. *Guide to LaTeX*. Tools and Techniques for Computer Typesetting, Addison-Wesley, 4th ed.

Krantz, Stephen G. 1997. *A Primer of Mathematical Writing*. American Mathematical Society.

Lamport, Leslie. 1994. *LaTeX: A Document Preparation System*. Addison-Wesley, 2nd ed.

Markey, Nicolas. 2005. Tame the BeaST: The B to X of BibTeX. Online. URL http://yum.math.hmc.edu/ctan/info/bibtex/tamethebeast/ttb_en.pdf. Accessed 2007 August 30.

> Excellent, in-depth introduction to BibTeX. Not only a good guide to day-to-day use, but also examines many of BibTeX's internals, which helps the reader understand *why* doing things one way rather than another is better.

Mathematics Subject Classification. 2010. Mathematics subject classification 2010. Website. URL http://msc2010.org/. Visited 2010 July 16.

Mittelbach, Frank, Michel Goossens, Johannes Braams, David Carlisle, Chris Rowley, Christine Detig, and Joachim Schrod. 2004. *The LaTeX Companion*. Tools and Techniques for Computer Typesetting, Boston: Addison-Wesley, 2nd ed.

Purrington, Colin. 2006. Sample scientific poster. Online; viewed 2006 April 25. URL http://www.swarthmore.edu/NatSci/cpurrin1/postertemplate.ppt.

> A PowerPoint template for a scientific poster, with information about the sorts of things that should and should not be included in such a poster.

———. 2011. Maintaining a laboratory notebook. Online; viewed 2011 September 14.

> Illustrated essay on how and why you should maintain a laboratory notebook. Includes useful information on selecting an appropriate notebook, as well as advice on pens that are insoluable in common laboratory and field solvents (water, methanol, etc.) based on experimentation.

———. 2016. Designing conference posters. Online; viewed 2016 August 25. URL http://colinpurrington.com/tips/poster-design.

> An excellent overview of how to create a poster for a scientific meeting. Also includes information on how to structure your presentation, how to dress, and other good ideas to help you and your poster be remembered.

Reynolds, Garr. 2005. Presentation Zen: The "Lessig Method" of presentation. Website. URL http://www.presentationzen.com/presentationzen/2005/10/the_lessig_meth.html.

———. 2011. *Presentation Zen: Simple Ideas on Presentation Design and Delivery*. Voices That Matter, New Riders, 2nd ed.

Swanson, Ellen, Arlene O'Sean, and Antoinette Schleyer. 1999. *Mathematics into Type*. Providence, RI: American Mathematical Society, updated ed. URL ftp://ftp.ams.org/pub/author-info/documentation/howto/mit-2.pdf.

> Not a LaTeX book, but one that talks about more fundamental questions of how to typeset (or even just type) mathematics. Topics such as breaking mathematical expressions, alternative ways of writing expressions, alignment, copyediting, and general writing advice as well. Available in PDF from the AMS (free).

Tufte, Edward R. 1983. *The Visual Display of Quantitative Information*. Cheshire, CT: Graphics Press.

———. 1990. *Envisioning Information*. Cheshire, CT: Graphics Press.

———. 1997. *Visual Explanations: Images and Quantities, Evidence and Narrative*. Cheshire, CT: Graphics Press.

———. 2006a. *Beautiful Evidence*. Cheshire, CT: Graphics Press.

———. 2006b. *The Cognitive Style of PowerPoint: Pitching Out Corrupts Within*. Cheshire, CT: Graphics Press, 2nd ed. See http://www.edwardtufte.com/bboard/q-and-a-fetch-msg?msg_id=0001yB&topic_id=1 for a discussion thread about this essay on Tufte's website.

Turing, Alan M. 1936. On computable numbers, with an application to the *Entscheidungsproblem*. *Procedings of the London Mathematical Society* 42(2):230–265.

TWG-TDS. 2004. A directory structure for TEX files. Website. URL http://tug.org/tds/.

University of Chicago Press, ed. 2003. *The Chicago Manual of Style*. Chicago, IL: University of Chicago Press, 15th ed.

The standard American work on typesetting style etc. (CHe)

Walsh, Norman. 2002. Help on BIBTEX names. Website. URL http://nwalsh.com/tex/texhelp/bibtx-23.html.