



*Computer
Science
Handbook*

Department of Computer Science
Gettysburg
COLLEGE

Computer Science Handbook

Department of Computer Science
Campus Box 402
Glatfelter Hall, 2nd Floor
Gettysburg College
300 North Washington St.
Gettysburg, Pennsylvania, 17325, USA

May 18, 2012

Contents

1	Introduction	2
2	Careers in Computer Science	3
3	Program in Computer Science	3
3.1	The Bachelor of Arts Program	3
3.2	The Bachelor of Science Program	4
3.3	Minor in Computer Science	6
3.4	Advanced Self-Placement	6
3.5	Prerequisite and Non-Major Courses	6
3.6	Grade Requirements	6
3.7	Department Honors	7
3.8	Current Course Offerings	7
3.9	Tentative Scheduling Pattern	7
3.10	Typical Sequence	7
4	The Faculty	9
5	Facilities	10

6	Life Beyond the Classroom	11
7	Student Projects	13
8	Student Internships	16
9	Graduate School Preparation	16
9.1	Recommendation Etiquette	18
10	Computer Science Fellowships & Scholarships	19
A	Checksheet	21
B	Planning Sheet	24

1 Introduction

There are some who would maintain that, since computer science relies on the use of machinery, it belongs more to the engineering discipline and has no place in the curriculum of a liberal arts college. What makes computer science a liberal arts discipline is not the machinery, but the process by which algorithms and data structures are devised to facilitate the use of the machinery in the solution of problems. It is this process that embodies the elements of a liberal education. There can be no doubt that it involves logical, precise thinking. It is a subject matter that relies on the major landmarks of human achievement that have clearly shaped our present and will have an increasing role in shaping our future. It is a carefully structured discipline that relies on its interactions with other disciplines for its very life blood. Computer science has made contributions that have changed the way other disciplines regard themselves and conduct their inquiries, and it has drawn on many other disciplines for making advances in its own base of information.

Our view of computer science at Gettysburg is that it is a study of the solution of problems through the use of algorithms and data structures and the implementation of these solutions on computing machinery. To the extent that we study the computing machine itself, it is a study of the logical design of the machine and the higher-order problems associated with having that machine operate in a problem solving environment. In short, we have clearly chosen to emphasize the algorithmic side of the computer science discipline. This side embodies the liberal arts tradition of Gettysburg College.

2 Careers in Computer Science

Gettysburg's computer science program can prepare you for any career where clear and logical thinking are valued. Our courses contain the technical aspects of computer science, but they also give equal attention to helping students communicate clearly and learn the independence necessary to carry an algorithmic way of thought into their future work, whether it be graduate study, a related discipline, or any career that values critical thinking.

Our graduates pursue a variety of exciting career trajectories. Some work for well known companies like Bell Atlantic, Hewlett-Packard, Prudential, IBM, Lockheed-Martin, and Perfect Order. Still others were double majors and have careers which focus on their other major; their computer science skills help them perform better rather than serving as the focus of their work. Some continue study at graduate school. Gettysburg computer science graduates have enrolled at Duke, Stanford, Dartmouth, George Mason University, University of Delaware, West Virginia, and Rochester Institute of Technology. Three graduates are professors at Gettysburg College, Williams College, and Washington and Lee University.

For specific information about career opportunities in computer science, talk with one of the computer science professors. For more specific advice concerning a career track in industry, we recommend meeting with Vice President of Information Technology and Professor of Computer Science Rodney ("Rod") Tosten. Rod has helped many students explore career opportunities through cutting edge work in Gettysburg College IT, as well as through his generosity with his real-world business experience.

3 Program in Computer Science

3.1 The Bachelor of Arts Program

The requirements for a B.A. in computer science are ten courses in computer science at the level of CS 107 or above. One of these courses may be selected from a list of approved courses in other departments.

The ten courses must include:

- CS 111 - Introduction to Computer Science I, or CS 107 - Introduction to Scientific Computation
- CS 112 - Introduction to Computer Science II

- CS 201 - The Mathematics of Computation
- CS 216 - Data Structures
- CS 221 - Computer Organization and Assembly Languages Programming
- CS 301 - Theory of Computation
- CS 440 - Advanced System Design (capstone course taken during the senior year)
- Plus three courses at the 300-level or 2 courses and one of the following: CS/Bio 251, Math 351, 366, Phy 240, Psych 315,316

In addition to the ten courses in computer science, a student must take:

- Math 111 - Calculus I (or its equivalent)

The course prerequisite structure is shown in Figure 1.

It is recommended that Math 111 be taken during the first year and CS 201 be taken no later than the third year.

Students who plan to find employment as a computer scientist or attend graduate school in computer science should consider taking Math 112, 212, and 366.

Students intending to do graduate study in computer science are advised to take the above plus Math 211 and 351, Physics 240 (Electronics), and further upper-level computer science courses. Further guidance for graduate school preparation is in Section 9 of this handbook.

3.2 The Bachelor of Science Program

A Bachelor of Science major in Computer Science has the same requirements as a Bachelor of Arts major in Computer Science plus four additional formal and/or natural science courses. At most two of the additional courses may be at the 100-level.

Formal sciences at Gettysburg College include Computer Science and Mathematics. Natural sciences at Gettysburg College include Biology, Chemistry, Environmental Science, Health Science, and Physics.

As with the B.A., CS 103 does not count towards B.S. degree requirements.

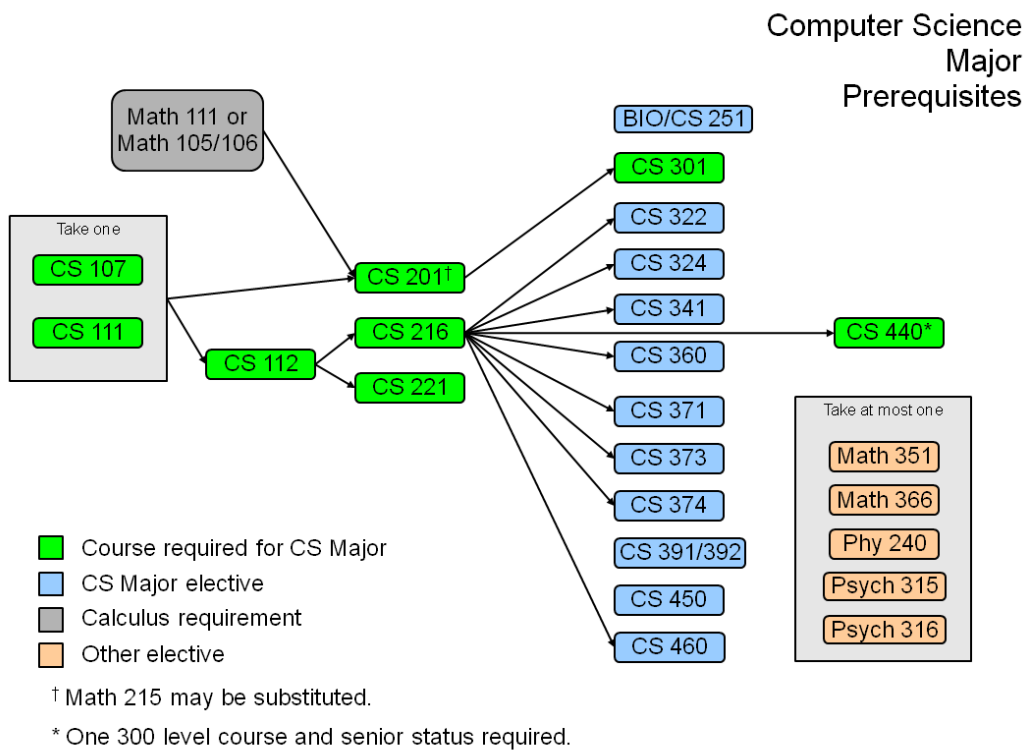


Figure 1: Course prerequisite structure.

3.3 Minor in Computer Science

A minor in computer science consists of six computer science courses numbered 107 or above.

3.4 Advanced Self-Placement

Students may opt for an advanced self-placement beginning with CS 112 or CS 216, skipping prior courses in the introductory sequence (which includes 107/111, 112, 216), with faculty approval. However, such course requirements are thereafter fulfilled by CS elective courses.

Example: Stu Dent self-placed into CS 112. He is exempted from the CS 107/111 prerequisite for the course, but will fulfill the CS 107/111 component of the major requirements by completing one additional CS elective. Had he self-placed into CS 216, he would need to complete two additional CS elective courses.

3.5 Prerequisite and Non-Major Courses

A student completing a Computer Science major course may not concurrently or subsequently be enrolled in a prerequisite, alternate prerequisite, or non-major Computer Science course. For example,

- 216 disallows 103, 107, 111, 112.
- 112 disallows 103, 107, 111.
- 111 disallows 103, 107.
- 107 disallows 103, 111.

3.6 Grade Requirements

All courses taken to satisfy the requirements for the B.A./B.S. degree or for the minor must be taken using the A-F grading system. To advance to a course with prerequisites, a minimum grade of C- is required for each prerequisite course.

3.7 Department Honors

Department Honors in computer science require participation in the co-curricular activities of the department, an overall grade point average of at least 3.0, and a computer science grade point average of at least 3.5.

In addition to Department Honors, the department also recognizes student excellence with two awards.

Superior Scholarship in Computer Science: The faculty may, at their discretion, elect a senior computer science major for this honor who exhibits extracurricular scholarly activity or superior academic performance.

Outstanding Computer Science Student: The faculty may, at their discretion, elect an outstanding senior computer science major for this honor.

For each honor, a certificate and cash award are presented to the student and his/her name appears on a plaque located outside of Glatfelter 207.

3.8 Current Course Offerings

Current semester course offerings are listed in CNAV¹. From the “Find It Now” drag-down menu, select “Courses”. Under “Multiple Condition Search”, select Dept “Computer Science”, set the Term to the current term, and click the “Submit” button.

The Gettysburg College course catalog is available online².

3.9 Tentative Scheduling Pattern

Figure 2 shows our current tentative course scheduling pattern. While not a guarantee of future offerings, it can offer guidance in course planning. For example, if there are electives that are particularly important to you, you may wish to anticipate their next offering and build your academic plan accordingly.

3.10 Typical Sequence

One typical scheduling sequence for a computer science major is shown in Figure 3. This is merely one example. Many other sequences are possible. There

¹<https://cnav.gettysburg.edu/index.cfm>

²<http://www.gettysburg.edu/academics/catalog/>

Course	Last Offered	Tentatively Scheduled
CS103 Intro. to Computing	F12/S13	every semester
CS107 Intro. Scientific Computation	F12/S13	every semester
CS111 Computer Science I	F12/S13	every semester
CS112 Computer Science II	F12/S13	every semester
CS201 The Mathematics of Computation	F12	fall semesters
CS216 Data Structures	F12/S13	every semester
CS221 Computer Org./Assembly Lang.	S13	spring semesters
CS251 Intro. to Bioinformatics	S13	every two years
CS301 Theory of Computation	S13	spring semesters
CS322 Intro. to Computer Networks	S13	every two years
CS324 Principles of Operating Systems	F11	every two years
CS341 Survey of Programming Lang.	F11	every two years
CS360 Principles of Database Systems	F12	every two years
CS371 Intro. to Artificial Intelligence	F12	every two years
CS373 Computer Graphics	S13	every two years
CS374 Compilers	S12	every two years
CS391/392 Selected Topics	S12	as scheduling permits
CS440 Advanced Systems Design	F12	fall semesters

Figure 2: Tentative course scheduling pattern

Year	Fall	Spring
1	CS 111 Math 111	CS 112
2	CS 216 CS 201	CS 221 CS 301
3	CS elective	CS elective
4	CS 440	CS elective

Figure 3: Typical scheduling sequence

are a checklist and planning sheet in Appendices A and B for constructing your course schedule.

4 The Faculty

Ivaylo Ilinkin (Ph.D. University of Minnesota) Ivaylo’s research interests are in computational geometry, bioinformatics, computer graphics, and computer science education. He spends his spare time travelling, reading, and discovering new restaurants and culinary experiences.

Chuck Kann [D. Sc., The George Washington University] Chuck is currently developing life cycle software to assist the Navy to do allowancing using statistical analysis of demand and Readiness Based Sparing (RBS) models. His other interests are concurrency and distributed computing, programmer psychology, and teaching. Outside interests include history, especially history of the American Civil War and the history of Adams county, and visiting and hiking battlefields.

Sunghee Kim (Ph.D. University of Minnesota) Sunny’s research interests are visualization, computer graphics, applied perception, computer vision, and human computer interaction. Outside school, she enjoys trying new gadgets and reading science fiction and mysteries.

Todd Neller (Ph.D., Stanford University) Todd is the current department chair. His computer science interests include artificial intelligence, machine learning, game theory, optimization, search, game-tree search, and decision theory. Beyond computing he especially enjoys games, cooking, music, and hiking.

Clifton Presser (Ph.D., University of South Carolina) Clif's primary research interests are scientific and information visualization. Beyond that, his interest in computing is very broad including web applications, virtual reality, image processing and artificial intelligence. Outside of the classroom Clif enjoys spending time with his family, reading and tinkering with his computers.

Rodney Tosten (Ph.D., George Mason University) Rod's special interests are in parallel processing, software engineering, and web applications. He is active in all aspects of the computer science curriculum in and out of the classroom. Rod's outside interests include model railroading, antique cars, archery, motorcycles, and woodworking.

5 Facilities

Information Technology maintains a campus-wide computing network. We support integrated development environments for several programming languages, and students have free access to a number of application packages. In addition, our department has its own state-of-the-art laboratory containing 17 Linux machines with 4GB RAM and dual-core processors. These machines are connected to a Sun Blade with dual UltraSPARC processors that is used as a local file server. These machines are used for some of the upper-division courses and for selected research topics. There are also several additional Linux machines clustered in other locations throughout the departmental area for student use.

Our laboratory is also equipped with a state-of-the-art projection system that includes four projectors and a SMART Board. There are two machines, one Linux and one Windows, available to the instructor or student assistants. The room is designed so that students can easily see what is being projected without having to turn away from his or her terminal. The furniture is also unique in that each workstation has a desk to work on that faces forward and a computer that faces sideways. The desks can be folded down to facilitate group work.

Additionally, we have:

- three MacBook Pro computers with Apple Developer Licenses, enabling our students to develop iPhone/iPod/iPad apps; iPad and iPod touches for testing
- SmartTable

- dozens of Arduino microcontrollers with soldering and prototyping materials and equipment (think Make magazine)
- Logic Pro with multiple third-party plugins, Axiom midi keyboard, condenser mic, and analog-to-digital input, for game music composition and sound effects design
- Canon EOS Rebel XT digital SLR camera, macro lens, tripod, and Photoshop for development of graphical elements
- an energetic faculty and student body to explore the possibilities of these resources!

6 Life Beyond the Classroom

A major in computer science should go beyond the required classes. You need to expose yourself to the broader cultures of the discipline as they are to be found in meeting a variety of professionals in the field, in trying your hand at challenging research problems, in reading the periodical literature, in building your own network among students with whom, if you stay in this discipline, you will cross paths the rest of your life. Here are some of the possibilities for involvement.

Colloquia and seminars: Colloquia are usually held during the Common Hour time slot. Topics include classical nuggets that do not fit into standard courses, special little problems that have caught the fancy of the speaker, an overview of some new development, information about graduate schools, applications from government or industry, a report on an honors project, and so on. All speakers are asked to address undergraduate majors. Speakers include visiting faculty from other schools, an occasional famous mathematician or computer scientist, our own faculty, people from the surrounding community, recent graduates, and current students.

While you may not follow all that is said in these talks, over a period of time, you will unconsciously learn new vocabulary, see unexpected connections, hear some idea that really does strike your fancy. You will come to know other people who participate in this kind of activity, and your presence will help generate that critical mass necessary to maintain a vigorous academic atmosphere.

Programming contests: Each year, the Association for Computing Machinery and several other groups sponsor a series of programming contests for undergraduates. In a programming contest, teams of three students are presented with six to ten problems to solve within a given time frame (usually three hours.) To get credit for a problem, the group must write a program to solve it and have that program approved by the judges.

Gettysburg started sending teams to some contests in 1991. During this time, our teams have had a very respectable record winning several local meets and placing in the top 20% in national and international meets. Most often, we compete locally with other colleges. However, we have also had enjoyable success at the ACM regional competitions, and University of Delaware's online competition.

Experience is not necessary; every student at the CS107 level or higher is invited to join us as we try to improve ourselves even further. Winning is always fun, but mostly we challenge ourselves with exciting problems which hone our problem-solving skills. Teamwork is often more important than programming skills, though obviously the latter is needed.

Department employment: There are opportunities for employment in the department and with Information Technology. We regularly hire graders, tutors, office assistants and in-class teaching assistants.

Research opportunities: Students completing the CS(111/107)-112-216 sequence may, in collaboration with faculty members, be involved in on-going research projects or study topics not covered by the regular course offerings. Such projects occur through independent study courses during the semester or in the summer. Contact any computer science professor if you are interested in their research or if you have found a topic that you would like to pursue.

Sources of information: The bulletin boards in the hall near the department offices are intended to carry items of interest to students. Headings mark areas for items related to faculty activities, student activities, computer science in the news, and job and internship information.

The place to look for current news in the department (a job announcement that just came in, a recent development in a search that is underway, a planning meeting, etc.) is on the bulletin boards next to Glatfelter 207.

We also keep files of graduate program announcements and information about fellowship and scholarships.

7 Student Projects

One of the best things about being part of a small department is that you get a chance to work on individual projects in close consultation with faculty members. Some of the more recent student projects include:

- Spending a summer working on configurations of parallel processors. The work resulted in a paper presented at the North American Transputers Users Group conference in Vancouver.
- Working on an independent study project on languages for parallel computation during a spring semester. A paper on that work was submitted for publication.
- Working on an independent study project concerning binary tree deletion during a semester. The work led to presentations in departmental colloquia, at the Moravian Student Mathematics Conference, and at the ACM's national meeting in Nashville.
- Spending a summer developing software for parallel processors. The work resulted in a paper presented at the World Transputer Congress in Harrogate, England.
- Spending a summer establishing the computer science backbone on the campus-wide network, converting our operating systems to Sun Solaris, and doing research on distributed processing. The work was presented as a paper to the International Conference on Parallel and Distributed Processing Techniques and Applications in Las Vegas.
- Spending part of a summer working on an application of computing in group theory. A factorization technique was discovered and a paper reporting this result was presented at a student conference.
- Spending a summer doing research on Java RMI in a parallel and distributed processing course. The work was published jointly with the sponsoring faculty member and presented at the International Conference on Parallel & Distributed Processing Techniques & Applications in Las Vegas.

- Spending a summer doing research on Enterprise JavaBeans. The results of the research were published jointly with the sponsoring faculty member and presented at the International Conference on Parallel & Distributed Processing Techniques & Applications in Las Vegas.
- Spending a summer and fall semester creating a Java-based netphone. Some of the major areas of research included SIP (Session Initiation Protocol), RTP (Real Time Protocol), and Java Media Framework. The results of the project were published jointly and presented with the sponsoring faculty member at the International Conference on Parallel and Distributed Processing Techniques & Applications in Las Vegas.
- *Java Smart Card Technology in the Classroom.* We investigated the use of embedded technology using Smart Cards within an undergraduate Computer Organization course. Smart Cards are credit card size devices that contain a computer processor and memory. Within the course, students were taught the Smart Card architecture and the Java programming interface that accompanies it.
- *Servlets and JSP in an Undergraduate Database Course.* In recent years database technology has become highly integrated with web applications. It is difficult to discuss or teach one without some reference to the other. The work we did demonstrates the inclusion of web related technologies into an undergraduate database course. Specifically, Java Servlets and Java Server Pages are employed to demonstrate web access of databases. We investigated the underlying technology, changes made to the course curriculum and details of the assignments and projects from the course.
- *Visualization of Distributed Applications.* We are currently investigating ways to communicate and display information that can be useful in building and debugging software applications that communicate amongst multiple computers. Our research combines the ability to perform “post-mortem” as well as run time analysis on a distributed application in order to determine what is going on between processes and when.
- *Clue Enhanced Detective Notepad project.* We developed the most advanced artificial intelligence (AI) for reasoning about the board game Clue. Whereas previous AI implementations are restricted to simple logical reasoning, our implementation uses modern sampling techniques to estimate the probability of any given card’s location.

A student designed and implemented the Visual C++ graphical user interface, and explores various means of clearly communicating mixed logical and probabilistic information.

Another student designed, implemented, and tested various sampling techniques for estimating probabilities with cardinality constraints.

- *Generation and Solution of Cluedoku Puzzles.* A student worked on the development of a deductive logic puzzle we call Cluedoku, based on the board game Clue. The student's work culminated in a system that automatically generates and rates the difficulty of Cluedoku puzzles. Not only did this work and our puzzle solving experience provide a variety of interesting puzzles, but it also revealed that advanced Clue reasoning is *hitting set reasoning*.
- *Analysis of the Bluffing Dice Game Dudo.* Dudo is a bluffing dice game originating with the Incan empire prior to the 16th century, popular as a folk game throughout South America, and currently enjoying international popularity in many variations such as Liar's Dice/Bluff, Perudo, and Cachito. Though played for centuries, one of our students was the first to collaborate in the computation of optimal mixed strategies for endgame play. Another student collaborated on the computation of the first approximation of optimal play for the complete 2-player game.
- *Generation of Rook Jumping Mazes.* In the fall of 2009, Gettysburg College faculty and students participated in a Rook Jumping Maze (RJM) design project, experimenting with maze generation and working to develop measures of RJM quality to aid fully-automatic computer generation of these mazes. Our results can be seen through the Rook Jumping Maze of the Day, a paper presented at the 7th International Conference on Computers and Games (CG2010) in Kanazawa, Japan, and a free iPhone app. Learn more at the Rook Jumping Maze website.
- *BabyEye* The goal of BabyEye is to automatically screen for strabismus from a baby photograph. Strabismus is an eye defect, in which a lack of coordination between the eyes cause the eyes to look in two different directions and may result in an inability to achieve binocular vision. The project consists of two parts: eye feature detection using computer vision techniques and automatic screening.

Two students did an extensive literature review of face and eye detection algorithms and techniques before designing and implementing various stages of reliably finding face and eye features from a single photograph of a baby. The important eye features for automatic screening were eye area, eye tilt angle, iris center, and pupil center.

- *TeraShake* The goals of TeraShake project was to find answers to important questions about earthquake and its impact on the Southern San Andrea Fault. We used the results of a physics-based simulation of a magnitude 7.7 earthquake on the Southern California regions for visualization.

A student designed the visualization algorithm of the time-varying data series and, using Ruby and VTK (Visualization ToolKit), implemented the algorithm to answer the questions posed by scientists.

- *Distributed Process Visualization* A student worked on building a visual debugger to display runtime information of multiple distributed Java processes. The visualization was rendered as an interactive tree using Java3D.

8 Student Internships

Numerous internship opportunities are available. Our students have held internships with Vanguard, NIST, NASA, Sun Microsystems, IBM, Microsoft, the Department of Commerce, CNAV Systems, and Dynamicsoft. For additional information about internships, contact the department or the Internship Office. Application materials for an internship is available on the department website³.

9 Graduate School Preparation

If you are considering graduate school, you should make yourself aware of graduate school course requirements. Our major requirements are those we consider to be the absolute essentials for a good computer scientist. However, we do not require all of the courses that some graduate schools may desire.

This means that you are responsible to surf websites of graduate schools of potential interest, and plan your courses accordingly. Unfortunately, there is no uniform standard across computer science graduate schools. However, we can

³<http://cs.gettysburg.edu/resources/internship.doc>

offer some general guidance based on requirements/suggestions from a number of CS graduate programs.

For example, this is a sampling of some recommendations for mathematical preparation from various schools:

- “[Have] adequate mathematical background including freshman and sophomore calculus and at least two other courses (e.g., linear algebra, abstract algebra, analysis, statistics).” - Cornell University
- “Calculus I, II, . . . Linear Algebra, . . . Methods of Applied Statistics” - UMass - Amherst
- “Students also need a reasonable foundation in mathematics, including calculus and linear algebra.” - University of Chicago
- “We also require at least one year of college-level Calculus.” - University of Virginia
- “Calculus I . . . Calculus II . . . Linear Algebra” - University of Maryland - College Park
- “. . . Calculus and Discrete Mathematical Structures . . . Most of our incoming graduates will have considerably more mathematical background, including courses like Probability and Statistics, and Linear Algebra.” - Ohio State University
- “. . . basic mathematics including calculus, probability and statistics, and discrete mathematics” - Clemson University
- “. . . A substantial background in mathematics, especially in calculus . . . , linear algebra . . . , finite mathematics . . . , probability/combinatorics . . . , and numerical analysis Such background should include at least two semesters of calculus and one semester in each of the other areas.” - Rutgers University

Note that these are the various mathematical expectations of graduate school. There is also considerable variation in expectations for undergraduate computer science courses. You would be well advised to start thinking about the kind of graduate school you might apply to and plan your computer science electives accordingly.

Some research universities are more interested in your research experience than your GPA. (For an example, see Carnegie-Mellon's "Applying to Ph.D. Programs in Computer Science"⁴.) One advantage of Gettysburg College is that you do not have to compete with graduate students for research interaction with professors. You can gain significant undergraduate research experiences if you are hard working and plan ahead.

There are two common ways that our undergraduates gain research experience. The first is to do approximately ten weeks of summer research with a faculty member. This requires the faculty member to put together an application for your funding. It takes time to write the application, and there are two funding cycles per year.

Your best bet is to talk with potential research advisors at the beginning of the fall semester. The first applications have generally been due in mid-October. The results of these applications are usually available before the end of the semester. In the worst case, you then have ample time to make alternate summer plans. However, if the fall deadline is missed, the second spring funding cycle has results available shortly before the end of the spring semester. No one enjoys waiting until the summer has almost arrived to find out if there is funding.

Again and again, be forward-thinking. Anticipate what you will need to do to gain opportunities for which you'll be thankful for years to come.

9.1 Recommendation Etiquette

Most graduate schools and many fellowships require faculty letter(s) of recommendation. Here are some tips and guidelines for seeking recommendations:

- Clarify in your mind what is most important to communicate about yourself and carefully consider who can best express this from their experience with you. For example, some research universities are much more interested in your research experience than your GPA. A research advisor would then be able to write a better letter than the professor who gave you the highest grade.
- Ask potential recommendation writers at least four weeks in advance. Respectfully explain your thoughts with regard to the previous point and ask if they feel they could provide a good recommendation. Generally, if you've

⁴<http://www-2.cs.cmu.edu/%7Eharchol/gradschooltalk.pdf>

had a long commitment to your goals and considered the previous point well, the professor will be glad to recommend you.

- Carefully prepare the recommendation materials well in advance. Often this means registering for online recommendation systems, printing out and filling out parts of forms, and preparing envelopes for the recommendations to be mailed. It is surprisingly common for students not to read the instructions for recommendation letter submission, leaving student name and data blanks on forms for faculty to fill in. This does not make a good impression at any level.
- Provide the materials to recommenders with a clear set of due dates. Like you, faculty are scrambling to meet deadlines all the time. Tasks without deadlines tend to be deferred. A simple sheet of paper with a list of all school application deadlines is good to provide. It is also a good idea to remind faculty as a particular deadline approaches.

10 Computer Science Fellowships & Scholarships

Below are a list of some fellowships and scholarships targeted to computer science majors.

- **DOE Computational Science Graduate Fellowships:** The Department of Energy provides support for computational science graduate students. Fellowships provide payment of all tuition and fees, a yearly stipend of \$36,000, yearly conferences, \$5,000 academic allowance in the first fellowship year, \$1,000 academic allowance each renewed year, a 12-week research practicum, and is renewable up to four years. Program of study must include: computer science, mathematics, and scientific or engineering discipline. There will also be a practicum assignment required at DOE lab. Students must be planning full-time PhD study at a U.S. university and support is limited to four years. Applicants must be exceptional senior undergraduates or in their first or second year of graduate study. (<http://www.krellinst.org/csgf>)
- **Goldwater:** Sophomores and juniors interested in careers in math, the natural sciences, and engineering are eligible to apply for this scholarship that awards up to \$7,500 per year. Students must have a B average, stand in

the upper fourth of their class, be a US citizen, and be enrolled as a full-time student in one of the disciplines mentioned. Students must receive an institutional nomination. Institutional deadline is generally in December. (<http://www.act.org/goldwater>)

- **Microsoft Scholarships:** Awarded to recipients for one academic year and will cover 100% of the tuition as posted by the financial aid office of the institution. All recipients will be required to complete a salaried summer internship of 12 weeks at Microsoft Corp. in Redmond, WA. Students must be making satisfactory academic progress (3.0) and majoring in computer science/engineering or a related discipline like math or physics, with a demonstrated interest in computer science. Deadline is usually the end of January. (<http://careers.microsoft.com/careers/en/us/internships-scholarships.aspx>)
- **National Science Foundation (NSF) Graduate Research Fellowships:** Awarded to graduate students in the beginning stages of study in the fields of mathematics, science, or engineering. The 2012 awardees received three years of support, with a \$30,000 annual stipend, \$10,500 cost-of-education allowance to the institution (anticipated to increase to \$12,000), international research and professional development opportunities, and TeraGrid Supercomputer access. Information is usually made available in August and applications are usually due in November. (<http://www.nsfgrfp.org/>)
- **Xerox Technical Minority Scholarships:** Scholarships are available in the amount of \$1,000 - \$10,000. Applicants must have a 3.0 or better gpa and be U.S. citizens or visa-holding Permanent Residents of African American, Asian, Pacific Island, Native American, Native Alaskan, or Hispanic descent. They must be enrolled as a full-time undergraduate or graduate student in any of the following technical fields: Chemistry, Information Management, Computing & Software Systems, Material Science, Printing Management Science, Laser Optics, Physics, Material Science, or Engineering. Application are usually due in September. Further information is available at <http://www.xeroxstudentcareers.com/why-xerox/scholarship.aspx>.

Further fellowship and scholarship information and guidance is available on the college website.

A Checksheet

Gettysburg College
Department of Computer Science
Checksheet - B.A. Computer Science

Name _____ Graduation Date _____

Required Courses

Course	Term	Grade
CS 111/107	_____	_____
CS 112	_____	_____
CS 201/Math 215	_____	_____
CS 216	_____	_____
CS 221	_____	_____
CS 301	_____	_____
CS 440	_____	_____
Math 111	_____	_____

Plus three CS courses at the 300-level or 2 courses and one of the following:
CS/Bio 251, Math 351, 366, Phy 240, Psych 315, 316

Course	Term	Grade
_____	_____	_____
_____	_____	_____
_____	_____	_____

Notes:

Gettysburg College
Department of Computer Science
Checksheet - B.S. Computer Science

Name _____ Graduation Date _____

Required Courses

Course	Term	Grade
CS 111/107	_____	_____
CS 112	_____	_____
CS 201/Math 215	_____	_____
CS 216	_____	_____
CS 221	_____	_____
CS 301	_____	_____
CS 440	_____	_____
Math 111	_____	_____

Plus three CS courses at the 300-level or 2 courses and one of the following:
CS/Bio 251, Math 351, 366, Phy 240, Psych 315, 316

Course	Term	Grade
_____	_____	_____
_____	_____	_____
_____	_____	_____

(continued on next page)

Plus four formal and/or natural science courses from formal and/or natural science departments, at most two of which may be at the 100-level. Formal sciences include Computer Science and Mathematics. Natural sciences include Biology, Chemistry, Environmental Science, Health Science, and Physics.

Course	Term	Grade
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____

Notes:

B Planning Sheet

Name _____ Graduation Date _____

Year 1	Fall _____	Spring _____
	_____	_____
	_____	_____
	_____	_____
	_____	_____
	_____	_____
Year 2	Fall _____	Spring _____
	_____	_____
	_____	_____
	_____	_____
	_____	_____
	_____	_____
Year 3	Fall _____	Spring _____
	_____	_____
	_____	_____
	_____	_____
	_____	_____
	_____	_____
Year 4	Fall _____	Spring _____
	_____	_____
	_____	_____
	_____	_____
	_____	_____
	_____	_____