CMPSC 111
Introduction to Computer Science I
Spring 2017

Syllabus

Course Instructor

Dr. Gregory M. Kapfhammer
Office Location: Alden Hall 108
Office Phone: +1 814–332–2880
Email: gkapfham@allegheny.edu
Twitter: @GregKapfhammer
Web Site: http://www.cs.allegheny.edu/sites/gkapfham/

Instructor's Office Hours

- Monday, Wednesday, and Friday: 11:00 am–12:00 noon (15 minute time slots)
- Tuesday and Thursday: 10:00 am–11:00 am (15 minute time slots)
- Tuesday: 2:30 pm–3:30 pm (15 minute time slots)
- Friday: 4:00 pm–5:00 pm (15 minute time slots)

To schedule a meeting with me during my office hours, please visit my web site and click the "Schedule" link in the top right-hand corner. Now, you can browse my office hours or schedule an appointment by clicking the correct link and then reserving an open time slot. Students are also encouraged to post appropriate questions to a channel in Slack, which is available at https://CMPSC111Spring2017.slack.com/, and monitored by the instructor and the teaching assistants.

Course Meeting Schedule

Lecture, Discussion, and Group Work Session: Monday and Wednesday 9:00 am–9:50 am
Practical Session: Friday 9:00 am–9:50 am
Laboratory Session: Thursday:, 2:30 pm–4:20 pm
Final Examination: Thursday, May 4, 2017 at 9:00 am

Course Description

> An introduction to the principles of computer science with an emphasis on creative expression through the medium of a programming language. Participating in hands-on activities that often require teamwork, students learn the computational structures needed to solve problems and produce artifacts such as interactive games and computer-mediated art and music. Students also learn how to organize and document a program's source code so that it effectively communicates with the intended users and maintainers. Additionally, the introduction includes an overview of the discipline of computer science and computational thinking. One laboratory per week. Prerequisite: Knowledge of elementary algebra. Distribution Requirements: ME, SP.

Course Objectives

The process of implementing and evaluating correct and efficient software involves the application of many interesting theories, techniques, and tools. In addition to learning problem solving and computational thinking skills, this class will teach students how to use, design, implement, and test algorithms in an object-oriented programming language. Students will learn more about fundamental concepts such as data types, conditional logic, and iteration while also discovering how to use single-dimension, multi-dimensional, and extendible arrays and to implement graphical applications. Students will also gain hands-on experience in the use, design, implementation, and testing of software during the laboratory and practical sessions and a final project. Along with learning more about how to effectively work in a team of diverse software developers, students will enhance their ability to write and present ideas about software in a clear, concise, and compelling fashion. Students will also develop an understanding of the fascinating connections between computer science and other disciplines in the social and natural sciences and the humanities.

Performance Objectives

At the completion of this semester, students must have a strong grasp of the basics of the object-oriented programming paradigm and an ever-deepening knowledge of topics like conditional logic, iteration, recursion, exceptions, and graphics programming. Also, students should be able to handle many of the important, yet accidental, aspects of implementing programs in the Java language. That is, students should be comfortable with the use of Vim as an integrated development environment and understand both the purpose and use of shell environment variables such as the `CLASSPATH`. Students should have a toolkit of programming language constructs that they can use to respond to the challenges that they encounter during the development and evaluation of software. Finally, students should demonstrate the ability to use both in-person discussions and cutting-edge software tools to effectively communicate and collaborate with a group of diverse team members.

Required Textbook

Java Software Solutions: Foundations of Program Design. John Lewis and William Loftus, Eighth Edition, ISBN: 978–0133594959, 806 pages, 2015.
(References to the textbook are abbreviated as "JSS" on the course web site and the syllabus).

Students who want to improve their technical writing skills may consult the following books.

BUGS in Writing: A Guide to Debugging Your Prose. Lyn Dupré. Second Edition, ISBN–10: 020137921X, ISBN-13: 978–0201379211, 704 pages, 1998.

Writing for Computer Science. Justin Zobel. Second Edition, ISBN–10: 1852338024, ISBN–13: 978–1852338022, 270 pages, 2004.

Along with reading the required textbook, you may be asked to study additional articles from a wide variety of conference proceedings, journals, and the popular press.

Class Policies

Grading

The grade that a student receives in this class will be based on the following categories. All percentages are approximate and, if the need to do so presents itself, it is possible for the course instructor to change the assigned percentages during the academic semester.

Class Participation and Instructor Meetings	10%
Midterm Examination	15%
Final Examination	15%
Quizzes	10%
Laboratory Assignments	30%
Practical Assignments	10%
Final Project	10%

These grading categories have the following definitions:

- *Class Participation and Instructor Meetings*: All students are required to actively participate during all of the class sessions. Your participation will take forms such as answering questions about the required reading assignments, asking constructive questions of group members, giving presentations, and leading a discussion session. Furthermore, all students are required to meet with the course instructor during office hours for at least fifteen minutes during the Spring 2017 semester. These meetings must be scheduled through the instructor's reservation system and documented on a meeting record that you submit on the day of the final examination. You also must regularly participate in the discussions on the Slack channels for this course. A student may request feedback on and will receive a final grade for this category.

- *Quizzes*: The two quizzes will cover all of the material in their associated module(s). While the second quiz is not cumulative, it will assume that a student has a basic understanding of the material that was the focus of the previous quiz. The date for each of the quizzes will be announced at least one week in advance of the scheduled date. Unless prior arrangements are made with the course instructor, all students will be expected to take these two quizzes on the scheduled date and complete the quizzes in the stated period of time.

- *Midterm Examination*: The midterm is an hour-long cumulative test covering all of the material from the class, practical, and laboratory sessions, as outlined on the review sheet. Unless prior arrangements are made with the course instructor, all students will be expected to take this test on the scheduled date and complete the test in the stated period of time.

- *Final Examination*: The final examination is a three-hour cumulative test. By enrolling in this course, students agree that, unless there are severe extenuating circumstances, they will take the final examination at the date and time stated on the first page of the syllabus.

- *Laboratory Assignments*: These assignments invite students to explore different techniques for designing, implementing, evaluating, and documenting software solutions to challenging problems that often have a connection to real-world concerns. Many of the assignments will require students to conduct experiments and collect, analyze, and write about data sets. To best ensure that students are ready to develop software in both other classes at Allegheny and after graduation, students will complete assignments both on an individual basis and in teams. When teamwork is required, the instructor will often assign individuals to teams.

- *Final Project*: This project will furnish you with the description of a problem and ask you to design, implement, describe, and orally present a correct and carefully evaluated solution. Completion of the final project will require you to apply all of the knowledge and skills that you have acquired during the course of the semester to solve a problem and, whenever possible, make your solution and results publicly available in a free and open fashion.

Assignment Submission

All assignments will have a stated due date. Electronic versions of the practical, laboratory, and final project assignments must be submitted to the version control repository that the student creates at the start of the semester. Additionally, the printed version of the assignment is to be turned in at the beginning of the class on that due date; the printed materials must be dated and signed with the Honor Code pledge of the student(s) completing the work. Late assignments will be accepted for up to one week past the assigned due date with a 15% penalty. All of the late assignments must be turned in at the beginning of the session that is scheduled one week after the due date. Unless special arrangements are made with the instructor, no work will be accepted after the late deadline. For any assignment completed in a group, students must also turn in a one-page document that describes each group member's contribution to the submitted deliverables.

Course Attendance

It is mandatory for all students to attend all of the class, practical, and laboratory sessions. If, due to extenuating circumstances, you will not be able to attend a session, then, whenever possible, please see the instructor at least one week in advance to describe your situation. Students who miss more than five unexcused sessions will have their final grade in the course reduced by one letter grade. Students who miss more than ten of the aforementioned events will fail the course.

Use of Laboratory Facilities

Throughout the semester, we will investigate many different software tools that computer scientists use during the design, implementation, and evaluation of computer software. The course instructor and the department's systems administrator have invested a considerable amount of time to ensure that our laboratories support the completion of all of the assignments and projects. To this end, students are required to complete all of the laboratory and practical assignments and the final project while using the department's laboratory facilities. The course instructor and the systems administrator normally do not assist students in configuring their personal computers.

Class Preparation

In order to minimize confusion and maximize learning, students must invest time to prepare for the class discussions, lectures, and practical sessions. During the class periods, the course instructor will often pose challenging questions that could require group discussion, the creation of a program or data set, a vote on a thought-provoking issue, or a group presentation. Only students who have prepared for class by reading the assigned material and reviewing the current laboratory and practical assignments will be able to effectively participate in these discussions.

More importantly, only prepared students will be able to acquire the knowledge and skills that are needed to be successful in this course, subsequent courses, and the field of computer science. In order to help students remain organized and effectively prepare for classes, the course instructor will maintain a class schedule with reading assignments and presentation slides. During the class sessions students will also be required to download, use, and modify programs and data sets that are made available through means such as the course web site and a version control repository.

Seeking Assistance

Students who are struggling to understand the knowledge and skills developed in a class, laboratory, or practical session are encouraged to seek assistance from the course instructor, the teaching assistants, or the departmental tutors. Throughout the semester, students should, within the bounds of the Honor Code, ask and answer questions on the Slack site for our course; please request assistance from the instructor, teaching assistants, and tutors first through Slack before sending an email. Students who need the course instructor's assistance must schedule a meeting through his web site and come to the meeting with all of the details needed to discuss their question.

Using Email

Although we will primarily use Slack for class communication, I will sometimes use email to send announcements about important matters such as changes in the schedule. It is your responsibility to check your email at least once a day and to ensure that you can reliably send and receive emails. This class policy is based on the statement about the use of email that appears in *The Compass*, the College's student handbook; please see the instructor if you do not have this handbook.

Honor Code

The Academic Honor Program that governs the entire academic program at Allegheny College is described in the Allegheny Academic Bulletin. The Honor Program applies to all work that is submitted for academic credit or to meet non-credit requirements for graduation at Allegheny College. This includes all work assigned for this class (e.g., examinations, laboratory assignments, and the final project). All students who have enrolled in the College will work under the Honor Program. Each student who has matriculated at the College has acknowledged the following pledge:

> I hereby recognize and pledge to fulfill my responsibilities, as defined in the Honor Code, and to maintain the integrity of both myself and the College community as a whole.

It is understood that an important part of the learning process in any course, and particularly one in computer science, derives from thoughtful discussions with teachers and fellow students. Such dialogue is encouraged. However, it is necessary to distinguish carefully between the student who discusses the principles underlying a problem with others and the student who produces assignments that are identical to, or merely variations on, someone else's work. While it is acceptable for students in this class to discuss their programs, data sets, and reports with their classmates, deliverables that are nearly identical to the work of others will be taken as evidence of violating the Honor Code.

Disability Services

The Americans with Disabilities Act (ADA) is a federal anti-discrimination statute that provides comprehensive civil rights protection for persons with disabilities. Among other things, this legislation requires all students with disabilities be guaranteed a learning environment that provides for reasonable accommodation of their disabilities. Students with disabilities who believe they may need accommodations in this class are encouraged to contact Disability Services at 332–2898. Disability Services is part of the Learning Commons and is located in Pelletier Library. Please do this as soon as possible to ensure that approved accommodations are implemented in a timely fashion.

Welcome to an Adventure in Computer Science

In reference to software, Frederick P. Brooks, Jr. wrote in chapter one of *The Mythical Man Month*, "The magic of myth and legend has come true in our time." Software is a pervasive aspect of our society that changes how we think and act. Efficient and correct software also has the potential to positively influence the lives of many people. Moreover, the design, implementation, evaluation, and documentation of software are exciting and rewarding activities! At the start of this class, I invite you to pursue, with great enthusiasm and vigor, this adventure in computer science.