# CS 211/211H: Object Oriented Programming

## (Fall 2017)

**Contact Information**

Dr. Shvetha Soundararajan
Email: shvetha@gmu.edu
Office: Engineering 4436
Phone: 703-993-6219
Office Hours: See Piazza for office hours information

GTAs: See the course Piazza site for contact information.

Piazza is the central site for our announcements, documents repository, and discussion board. The announcements and discussion board are part of the *required reading* for the course.

All instructors and TAs can view all material on Piazza. **Do not** e-mail course staff about programming problems; use the discussion board. Use public posts on Piazza to discuss programming project requirements, labs, and other material related to the course. When prompted by a TA, use private posts on Piazza to share portions of your code pertaining to your questions. Don't share your project code in public posts. Refer to the Piazza main page for etiquette on what should be posted publicly versus privately. **Email** course staff only for **logistical issues** such as meeting outside of office hours, missing lab/lecture, grading disputes, medical situations, etc. Email addresses are listed on Piazza.

BlackBoard is used for project submission and to post grades. **Office Hours** are listed for all Instructors and TAs on Piazza.

**Course Outcomes**

- An understanding of basic object-oriented (OO) programming concepts and principles.
- An ability to apply basic object-oriented principles and techniques in the development of software systems using a specific programming language.
- An ability to effectively develop software systems using both basic command line tools and sophisticated integrated development environments, and to understand the

advantages and limitations of each.
- An ability to successfully perform debugging operations and techniques.
- An ability to perform software development in both individual and team environments.
- An understanding of programming-related references/resources available to software developers and the ability to use them effectively – both in ongoing projects and in the acquisition of new technical skills.
- An understanding of how acquired programming skills facilitate success in upper level CS courses and in various professional environments.

**Textbook**

*Recommended*: Building Java Programs, 3rd ed., by Reges and Stepp.

*Recommended*: Lab Manual (sections will be posted online on the course Piazza site for free download).

**Computing**

It is assumed you will have access to a computer with the ability to edit, compile, and run Java programs. Some university labs provide this ability. If you have difficulty accessing a suitable environment, contact the course staff.

**Coursework**

**1. Lectures**

During lectures we will discuss programming concepts and instructors will provide demos of programming relevant to other course work. Programming labs will be for students to work on additional exercises and get immediate help from teaching assistants. In addition to attending the regular meeting times, you are strongly encouraged to visit the professor and teaching assistant(s) during office hours to further your understanding of the material: we are here to help you learn.

**2. Textbook Readings**

Readings from the textbook relevant to each lecture are listed in the schedule. You will increase your understanding of lectures by reading associated textbook sections ahead of time, though this

is not assumed. We may provide additional reading material to supplement the textbook which will be posted on the course web page.

There are also several free online Java tutorials available that you can use.

**3. Labs**

Labs meet once per week and attendance is required. Labs roughly alternate between **Lab Exercises** and **Lab Quizzes/Tasks**.

Associated with lab are readings from the Lab Manual posted online. It is assumed that students read the scheduled lab manual sections prior to each lab; doing so will make it relatively easy to complete the labs during the allowed time. **This is one of our main sources of practice**. If you don't practice, you won't succeed.

**Lab Exercises**

About half of labs will involve Lab Exercises. These are short sets of programming problems which are designed to be completed during the lab time so long as students have kept up with reading and lectures. Teaching Assistants will give a brief introduction to the problems and then be present to assist as students work on the exercises.

**Lab exercises are open resource and open collaboration.** Students may freely discuss how to solve the exercises with anyone, examine each other's code, assist one another in debugging, and employ any online or physical resources to complete the exercises. No penalties will be assessed for similar looking code on Lab Exercises.

**Lab Quizzes and Tasks**

Some of the labs will involve either a Lab Quiz, a set of paper and pencil questions, or a Lab Task, a programming task on a computer. **Lab Quizzes and Tasks are closed resource, no collaboration allowed.** For Lab Tasks, students may use their own programming environment and submit their solutions to Blackboard by the end of the lab. No external resources may be used for Tasks.

Students will take the Quiz or Task in the lab room and be monitored by teaching assistants. All materials must be submitted by the end of the lab period according to the instructions associated with the assessment.

**4. Programming Projects**

Students will receive a number of programming projects during the semester. Each project will involve writing programs and answering questions about them to illustrate an understanding of course material.

Each programming project will have an "Honors Problem" which honor section (H01) students must complete. Other sections are not required to do these sections and will not receive any credit for completing them but are free to try. The normal projects will normally be scored out of 100 points, and earning 100 points would be a perfect score. The honors problem will be some additional amount of points. If a particular project had a 15 point honors problem, then they would need 115 points to get a perfect score. Disregard blackboard's "points possible" claims if it doesn't match your style of section.

**5. Exams**

There will be two midterm exams during the course during the regularly scheduled lecture time. There will also be a comprehensive final exam at the end of the semester. Refer to the schedule for dates of the exams.

**Grading Policy**

| | |
|---|---|
| Lab Exercises | 5% |
| Lab quizzes and tasks | 10% |
| Programming Projects | 40% |
| Midterm Exams (2) | 20% |
| Final Exam | 25% |

Contesting of grades on any/all submissions must be requested within one week of the item's return. No grade changes will be considered subsequent to that deadline, or after the final exam meeting.

Final grades are calculated on a 10-point scale per letter grade, with the upper and lower 2% of each 10% earning a + or -.

A+ (>= 98.0%) A (>= 92.0%) A- (>= 90.0%)
B+ (>= 88.0%) B (>= 82.0%) B- (>= 80.0%)
C+ (>= 78.0%) C (>= 72.0%) C- (>= 70.0%)
D (>= 60.0%)
F (< 60.0%)

**Note**: **Per departmental policy, you must pass a significant exam threshold to receive a passing grade in this class, regardless of your performance on other assignments. Failing the final exam, will result in a failing grade (F) for the entire course unless you have achieved an average exam score >=65%. This average score is calculated as a weighted average of all your exam scores (2 midterms and the final).**

**Lab Exercise Grading**

In order to receive credit for completing Lab Exercises do the following.

**You must attend your *assigned* lab section and be present for attendance.** Complete your program and pass the public tests that are provided. Credit will be assigned on the proportion of tests that are passed. Submit your exercises as per the instructions in each lab, usually zipping and uploading to Blackboard.

No late submissions for Lab Exercises will be accepted. Missing the deadline results in 0 credit. Failing to attend lab will result in a 0 for that week's lab exercises.

**Lab Quiz and Task Grading**

In order to receive credit for lab quizzes and tasks, you must be present in your assigned lab and submit your work by the end of the lab period. **The lowest Quiz or Task score for the semester will be dropped. No lab makeups will be given under any circumstances.** Plan to attend all labs so that in the event that you miss a quiz, the resulting 0 score is dropped.

**Project Grading**

**No collaboration is allowed on programming projects unless explicitly indicated in the project description.** Utilize the discussion board and office hours of course staff if you have questions about the projects.

Project grading will usually be divided into two portions.

1. Automatic Testing: Public tests will be used to assess the correctness of programs. Running these tests while constructing your program will indicate exactly what your score on this portion will be.

2. Manual Inspection: Each project will include a checklist of features your programs should exhibit. These usually comprise things that cannot be easily checked via unit tests such as good variable name selection, proper decomposition of a problem into multiple functions or cooperating objects, overall design elegance, and proper asymptotic complexity. These features will be checked by graders and assigned credit based on level of compliance.

The policies for project submission are as follows.

- Projects each have a deadline; no late work is accepted more than 48 hours after the deadline, and penalties for late work apply (see below). Work turned in more than 48 hours late doesn't count.
- Each student starts the semester with three Day-Late Tokens (not applicable for projects that are common to all sections). Whenever a student turns in a project late, tokens are **automatically** applied to the assignment.
- A "ceiling penalty" of 25% is assessed each 24-hour period entered after the deadline (when late- tokens are gone). For example, if you turn in work half a day late (one 24-hour period late), you can't earn higher than 75%.
- Work turned in more than **two 24-hour periods late will earn zero points** and will only be inspected as a courtesy (no grade assigned) as time permits. To clarify, you can't use tokens and/or penalties to turn work in 3 days late; this would earn zero points, because it has missed the 48- hour-late hard deadline.
- You can submit work to BlackBoard as many times as desired. Only the last submission will be graded, and all others are ignored. If you anticipate being rushed around the submission deadline, be sure to submit a version before the deadline as a backup, in case you find yourself only moments late in meeting a deadline, or your wifi goes out at the last moment.
- GTAs grade the programming projects for the lab sections they teach.
- On-time submissions will generally be graded and available a week from submission, though exceptional circumstances may cause delays. Late submissions will be graded in as timely a fashion as schedules allow.
- *Be sure you actually turned in your work!* We need .java files; .class files are worthless as a submission. If you tend to keep earlier versions in extra files, make sure you turned in

the final product, as that is what will be graded.

- **BACK UP** your work, always! Without fail, each semester multiple students will lose, drown, crack, reformat, magnetize, or generally destroy their computers at the worst possible time. Keep everything somewhere backed up, such as a DropBox or Google Drive folder, so that this doesn't happen to you.

**Honor Code**

All students are expected to abide by the GMU Honor Code and the CS Department's Honor Code policies. This policy is rigorously enforced. All class-related assignments are considered individual efforts unless explicitly expressed otherwise (in writing). Review the university and department honor code and present any questions regarding the policies to instructor. Cheating on any assignment will be prosecuted and result in a notification of the Honor Committee as outlined in the GMU Honor Code.

**Disability Accommodations**

Students with a learning disability or other condition (documented with GMU Office of Disability Services) that may impact academic performance should speak with me ASAP to discuss accommodations.