

CS 211: OBJECT ORIENTED PROGRAMMING

GEORGE MASON UNIVERSITY DEPARTMENT OF COMPUTER SCIENCE
SUMMER 2019

1. COURSE BASICS

INSTRUCTORS:

NAME: Michael Neary
EMAIL: mpneary_at_gmu_dot_edu
HOURS: 2:30-4:30pm Thu or [by appointment](#)
OFFICE: 4417 Engineering

GRADUATE TA:

Sabiha Salma (ssalma_at_gmu_dot_edu)

PREREQUISITES: CS112 (C or better) and access to a Java-capable computer

FORMAT: Lecture plus twice weekly lab

LECTURES:

Section	Days	Times	Location	Instructor
001	Tue/Thu	10:30-1:10pm	129 Blueridge Hall	Neary

TEXTBOOKS:

	GMU	<i>CS 211 Lab Manual</i>	available for free download from https://cs.gmu.edu/~marks/211/textbook/
	zyBooks	<i>Programming in Java</i>	available online at https://learn.zybooks.com/ (access using the sign-in code GMUCS211NearySummer2019)
(Optional)	Reges and Stepp	<i>Building Java Programs</i> , 3rd ed.	available online at https://practiceit.cs.washington.edu/

PIAZZA: [Piazza](#) will be used for all official announcements and online discussion; any information discussed on Piazza will be assumed to be known to students.

- Course schedule, announcements discussion. GTA/UTA contacts and office hours will be on Piazza too. Sign up yourself by following this link: <https://piazza.com/gmu/spring2019/cs211/>
- 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.
- 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 above and on Piazza.

BLACKBOARD: [Blackboard](#) will be used for course slides, assignments (including submissions), and grades.

2. COURSE INFORMATION

COURSE DESCRIPTION: Thorough treatment of programming according to object-oriented principles. Introduces classes, interfaces, inheritance, polymorphism, and single dispatch as means to decompose problems. Covers intermediate programming techniques including error handling through exceptions, arrangement of source code into packages, and simple data structures. Intermediate debugging techniques and unit testing are covered.

OUTCOMES

1. An understanding of basic object-oriented (OO) programming concepts and principles.
2. An ability to apply basic object-oriented principles and techniques in the development of software systems using a specific programming language.
3. 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.
4. An ability to successfully perform debugging operations and techniques.
5. An ability to perform software development in both individual and team environments.
6. 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.
7. An understanding of how acquired programming skills facilitate success in upper level CS courses and in various professional environments.

3. COURSEWORK

ZYBOOKS: online textbook with practice problems; weekly deadlines for completing the practices are checked automatically.

LABS:

- *Exercise:* programming assignment; open resources; collaboration/group work allowed; several days to submit.
- *Quiz:* pencil and paper; closed resources; individual; due same-day during lab.
- *Task:* computer assignment; closed resources; individual; due same-day during lab.

PROJECTS: programming assignment; individual work; possibly more than a week to complete.

EXAMS: individual work; in-class.

4. GRADING PROCEDURES

GRADE DISTRIBUTION

Material	Weight	Drop policy
zyBooks	5%	lowest 15 subsections
Labs (weekly)	15%	lowest three
Programming projects (5-6)	40%	lowest one
Midterm exam	15%	-
Final exam	25%	-

GRADING POLICIES

- Grades within a category (i.e. midterms, projects, labs) are weighted equally.
- Students must show their Mason ID card when taking exams.
- By department policy, the student must pass the final *or* the weighted average of both exams and your lab scores must be a passing grade (≥ 60) in order for a student to pass the course.
- The final exam is cumulative; a high final exam score dominates (replaces) lower scores on one or both of the midterms.
- Challenging of any grade must occur within a week of when the graded assignment has been returned.
- Any number of resubmissions are allowed (the most recent is used), however a resubmission turned in after the deadline will be considered a late submission.
- Lab attendance is mandatory; for lab exercises which do not need to be submitted during lab, attendance can impact one point (out of 10) of each assignment; a grace of up to 3 missed points is allowed (this is independent of dropped lab scores and late tokens).
- Lab exercise grading is fully automated. Quizzes may be hand-graded. Lab task and project grading is partially automated. Even when manually graded, code which does not compile will receive a zero in most cases.
- Absences are absorbed by the drop policy - in general, make-ups are not allowed except on exams (provided a valid excuse).

- Assignments are typically due at midnight on the listed due date.
- Late work can be mitigated by *emergency tokens* (see below).

EMERGENCY TOKENS

- Every student begins the semester with a non-replenishing allotment of four emergency tokens.
- For every emergency token, an assignment (programming projects and lab exercises) can be submitted a day (24 hours) late without penalty. An assignment can be submitted two days late at the cost of two tokens.
- No assignment will be accepted more than 2 days late (i.e. if 3 tokens are available, at most 2 of them can be used on the assignment).
- Without applying tokens, a one day late assignment can receive a maximum score of 75%, while a two day late assignment can receive a maximum score of 50%. Beyond that an assignment receives a zero, regardless of tokens. The score is a score cap rather than a penalty (a one day late assignment which would have received an 85% would still get a 75%).
- Tokens are applied in the order that assignments are submitted, such that projects have priority over labs (tokens are first applied to projects and then to labs if there are any remaining); a student cannot pick and choose where to use tokens to maximize impact.
- At the end of the semester, each unused token is worth 0.25% extra credit towards your final grade. This makes for a maximum of 1% extra credit if you don't use any emergency token.

GRADING SCALE

Grade	A+	A	A-	B+	B	B-	C+	C	C-	D	F
max	↑	97	91	89	87	81	79	77	71	69	59
min	98	92	90	88	82	80	78	72	70	60	↓

TIPS

- *Make backups*, because the unexpected happens, and cannot be used as an excuse to get an extension.
- Submission times are automatically recorded by Blackboard, and there's no distinction between a tiny bit late and nearly a day late - plan ahead to make sure that your submission is on time.
- To receive a grade, the submission must be gradable. This means submitting `.java` source rather than compiled `.class` files or word documents containing the source. It also means that the code must be submitted on Blackboard rather than simply saved.

5. HONOR CODE

Unless specific instructions are given to the contrary, programming assignments are an individual effort, no group work is allowed. In addition to code, this includes the sharing of test cases, pseudocode, or approaches, receiving assistance in debugging code, as well as the use of external Internet sites.

Both the [GMU Honor Code](#) and the [CS Department Honor Code](#) apply in this class. Any use of a direct contribution on any program, homework, quiz, or exam will be reported as a violation of the honor code.

6. SPECIAL ACCOMMODATIONS

Students who have a right to accommodations due to disabilities or other conditions should discuss this with the instructor as soon as possible. Accommodations will follow the recommendations of the University's [Office of Disability Services](#).