ECE 39595 Object Oriented Programming (with C++ and some Java)

Fall 2022

Sam Midkiff

smidkiff@purdue.edu

Course Staff

- Instructor: Professor Sam Midkiff, EE 310 (up the center stairs in the EE building, take a left, follow the hallway to the gray double doors, and my office is on the left).
- Teaching Assistants:
  - David Burnet
  - Fernando Davis Rivero

About myself

- Professor at Purdue since 2002
- Research in Compilers and Parallel Programming
- Have taught this course or its predecessor for many years
- I also teach a graduate course on parallel programming

About this offering of the course

- Two parallel offerings with same title and name:
  - One offered by Prof. Sanjay Rao
  - This section.
- The two sections are VERY DIFFERENT with completely different
  - Exams
  - Projects
  - Grading scheme
- Material will overlap, but can be quite different and will be sequenced differently.
- Please treat two sections as two independent courses.
  - *You cannot attend lectures of one section assuming you will get same content to the other lecture.*

Why two parallel offerings?

- Major growth in course enrollment since originally created
  - About 200+ now (vs. 50 in initial years).
- Recognition that course has to be redesigned with class scale in mind
- Move towards
  - Multiple auto-gradable projects instead of one large demo-based project
- Difficult to do this redesign with the full 200+ enrollment
- Our decision:
  - Larger section taught this year by me (similar structure to previous years)
  - Smaller section taught by Prof. Rao (will become default structure over time)

The syllabus and grading.

- Read the syllabus.
- Guaranteed cut-off points, i.e., you make in this range and you are guaranteed to make at least this grade
  - 90 and above, a A
  - 80 and above, a B
  - 70 and above, a C
  - 60 and above, a D
  - Below 60, an F
- The cut-offs usually run a little lower
- I usually do some +/- grading. This involves taking the highest Bs and making them A-, making a few remaining top Bs into B+, and similarly for Cs and Ds.
- These cut-offs are different than in the other section

Components of your final grade

- What goes into your grade
  - Three tests, worth 30% of your grade (each is 10%)
  - There is no final
  - The project is worth 40% of your grade
    - The first 2 project steps are worth 5% each of your grade.
    - The final project turn-in will be worth 30% of your grade, for a total of 40 points.
    - If you are happy with the grade on step 2, don't turn in step 3 and you'll get 6*(step2 grade) points for the final project turn-in
  - Homework will be worth 30% of your grade. Currently I have 12 C++ homeworks and 3 Java homeworks to be assigned. If I keep to that, I will drop 3 C++ homework grades and one Java homework grades.
- Again, this is different than in the other section.

Content for this section

- Primary focus: C++, object-oriented programming concepts, and some Java
- Less focus on memory management than the other section, and more focus on C++ features than the other section
- Less emphasis on data structures fundamentals and more emphasis on OO Design Patterns and Java than the other section

Please consider switching to other section if topic coverage and emphasis don’t work for you.

General information

- If you send me email put “39595” in the subject. This will increase the odds of your email being read, and being found a day later if I don’t get to it immediately. I reserve the right to ignore email without this in the subject line.
- TA and office hour information is available on the syllabus on Brightspace and as a pinned post on piazza.
- We will use Piazza to answer questions. Homework, homework solutions, old tests, etc., will be posted on Brightspace
    - Ask technical questions on Piazza – we tend to have good turn-around on answering these.
    - If I get a non-personal question by email I will post the answer to Piazza.
- Assignment grades will be posted on Brightspace
    - Brightspace attempts to assign a total percentage based on these – don’t believe it. For this to be correct I have to enter a grading scheme into Brightspace, and I would rather not spend any of what remains in my life figuring out how to drop low grades, etc., in Brightspace.
- MW office hours will be in person and remote. TuTh will be remote. The zoom link is in the syllabus.
- Lectures will be recorded. If you watch the lecture live, I will try and hang around online for a while after the lecture to answer questions. And you can ask questions during the lecture.
- Project and homework questions related to programming should be directed to the TAs first. If they cannot answer them, come to me.

The syllabus and grading.

- Read the syllabus.
- Guaranteed cut-off points, i.e., you make in this range and you are guaranteed to make at least this grade
  - 90 and above, a A
  - 80 and above, a B
  - 70 and above, a C
  - 60 and above, a D
  - Below 60, an F
- The cut-offs usually run a little lower

Plus/minus grades and other grade policies

- I usually do some +/- grading. This involves taking the highest Bs and making them A-, making a few remaining top Bs into B+, and similarly for Cs and Ds.
- What goes into your grade
  - Three tests, worth 30% of your grade (each is 10%)
  - There is no final
  - The project is worth 40% of your grade
    - The first 2 project steps are worth 5% each of your grade.
    - The final project turn-in will be worth 30% of your grade, for a total of 40 points.
    - If you are happy with the grade on step 2, don't turn in step 3 and you'll get 6*(step2 grade) points for the final project turn-in
  - Homework will be worth 30% of your grade. Currently I have 12 C++ homeworks and 3 Java homeworks to be assigned. If I keep to that, I will drop 2 C++ homework grades and one Java homework grades.

For policies, including grades, the syllabus is the ground truth

- One principle we'll discuss in class is that when information is duplicated, one copy will be wrong because it doesn't get updated, etc.
    - I may post summaries of information from the syllabus in a prominent place on the course website, etc. If it disagrees with the syllabus, believe the *latest* copy of the syllabus.
    - This includes what is on these slides
- If I change due dates they will almost always be changed to a later date.

Cheating

- Cheating will be punished severely
  - At a minimum you will fail the assignment
  - At a maximum you will fail the course
  - The decision will be mine, and mine alone
- All cases of cheating will be reported to the Dean of Students and the ECE Department
- If it’s getting close to a deadline and it’s late at night and you are desperate, and considering cheating
  - Don’t do it
  - Talk to me

Collaboration and Submitting work

- You can discuss lecture, homework, lab, or programming assignments with anyone. You should not share homework code. You can share project **code *only*** with your project partner (if you have one).
- You can have one and at most one partner the project.

- Having code similar to another project's code may lead to an F in the assignment and/or the course.
- See each assignment and the syllabus for submission instructions. Your programs should compile and run using g++ from a command line in Linux. Note that Microsoft VisualStudio will let you put Microsoft extensions into your code that will not compiler with g++! For the Java homework, make sure you can compile and run with javac and java from the command line.

Some additional info

- If you haven’t had 264 or its equivalent, send me email
- If you don’t understand pointers, send me email.
- You should be able to write and compile programs
- Compilers are for catching syntax errors. I won’t test on syntax – syntax isn’t fundamental and I hope we’ll program enough and you’ll see enough code that you can figure out what is syntactically incorrect when the compiler shows you where to look.

One part of the first homework: use an IDE

- Interactive development environments (IDEs) are useful for writing OO programs
- For the first programming assignment you will get access to and use an IDE
- Xcode in OSX, Visual Studio in Microsoft, Eclipse for anything including Linux, and many others
    - Use what you want, but use and IDE
- An IDE is necessary to use a debugger in Java