

# **COP 3330 Syllabus**

## **Object Oriented Programming – Fall 2026**

### **Section 1**

**Course Prerequisite:** COP 2500 or Placement Test

**Class Time:** MW 12:00 – 1:15 pm

**Class Location:** HS1-119

**Course Web Page:** <https://www.cs.ucf.edu/courses/cop3330/fall2026>

**Instructor:** Arup Guha

**Office:** HEC – 240

**Email:** [dmario@ucf.edu](mailto:dmario@ucf.edu)

**Office Hours:** Listed at <https://www.cs.ucf.edu/~dmario/ucf/OH.html>

**Course Description:** COP 3330 provides an introduction to object-oriented programming via the Java programming language for those with *the equivalent of one prior class of programming experience*.

We will cover the following topics in Java: input, output, variables, arithmetic expressions, if statements, loops, loop control, nested loops, static methods, arrays(1D and 2D), using the following built in classes: String, Random, ArrayList, Stack, Queue, HashSet, HashMap, PriorityQueue, Classes, Inheritance, Interfaces, Abstract Classes, Polymorphism, Exception Handling, File I/O and GUIs in Java.

**Recommended Book(s):** Any book that covers the basics of Java will do. Many students who succeed in the course never buy a book because online resources for both languages are excellent. If you prefer to have a book, please feel free to ask me if a particular book would be suitable as a supplement for our course. The reality is that when a course textbook existed, reading it did not correlate with the course grade **at all.** **Writing extra programs did though.**

## Other Important Course Policies

1) Cheating will not be tolerated. **If a student is caught cheating, then the grade on that assignment for all students knowingly involved (the person providing answers as well as the one taking the answers) will be a -25%.** (Note, this is less than 0%.) Since discussion of concepts with other students is often helpful, cheating must be more clearly defined. In particular, the following items are cheating: **getting any code directly from any generative AI, even if it's modified afterwards**, copying a segment of code of three lines or more from another student from a printout or by looking at their computer screen, taking a copy of another student's work and then editing that copy, and sitting side by side while writing code for assignments and working together on segments of code. In all of these situations, **BOTH people responsible**, the one from whom the three lines of code are taken as well as the person who takes those lines of code are engaging in academic misconduct. For example, if someone makes an electronic copy of their code accessible to ANYONE in the class (except for themselves) before 48 hours after an assignment is due, they are automatically culpable of academic misconduct. It does not matter if the recipient of the code doesn't use it, uses it a little, or copies it directly.

**If you get stuck on an assignment, please ask either a TA or me for help instead of getting help from another student.** Part of the learning process in programming involves debugging on your own. In our experience, when a student helps another student with an assignment, they rarely allow the student getting help to "figure out" problems on their own. Ultimately, this results in a lack of debugging experience for the student receiving help. The goal of the TAs and instructors is to provide the facilitation necessary for students to debug and fix their own programs rather than simply solving their problems.

**You are encouraged to work together on ungraded exercises and use those as an avenue to improve your understanding of concepts.**

2) In order to take a make-up exam, you must request one from the instructor. The instructor will grant requests using his own judgment by applying the following general rule: "Make-up exams will only be given if the reason for missing the exam was out of the student's control." For example, being hospitalized unexpectedly is out of a student's control, but oversleeping or going to happy hour is not out of a student's control. ***If possible, it is recommended that the instructor be contacted before the exam. Specifically, make ups will not be given if there's suspicion that the student was unaware of the exam before it occurred.***

3) **NO LATE ASSIGNMENTS WILL BE ACCEPTED.** If a work/military/family/personal reason prevents you from turning in an assignment on time, please contact me as early as possible and I will make alternate arrangements. If the situation is serious enough, I may suggest pursuing an Incomplete or dropping the course.

4) Both the course web page and WebCourses will be crucial elements of the course. ***It is your responsibility to check both of these before every class meeting for any updates that may be posted.*** Some clarifications may only be given in class and won't be posted online at all, so make sure you keep up with announcements in class.

## Grading

The final letter grade will be based upon the items listed below. Plus/minus grades will be issued, when deemed appropriate.

| Item                                       | Percentage   |
|--------------------------------------------|--------------|
| Individual Homework Assignments (10 total) | 30 (3% each) |
| Exam 1                                     | 20           |
| Exam 2                                     | 20           |
| Final Exam                                 | 30           |

**In order to pass the class you must earn at least 40% on the final exam.** (Thus, if you have 75% in the course but earn 30% on the final, you still get a C- in the course even though your percentage may qualify for a B.)

### *Letter Grades*

I assign letter grades a bit differently than other professors. I do not use a straight 90-100, 80-90, etc. grading scale. Rather, at the end of the semester, I chose my lines for each grade. The drawback to this technique is that students do not know exactly what letter grade they are earning during the semester. The advantage to it is that if I make a difficult quiz/exam, I can adjust my grades accordingly, so students don't get punished for my exam making skills. In the past, my A line has ranged from about 83-87%, my B line has ranged from 67-70%, and my C line typically ranges from 52-55%. I do not guarantee that these will be accurate for this semester, but I wanted to give you a rough idea of how the grades have gone in the past. After each exam, I will update WebCourses to show you what letter grade you have at that point in time. If you have further questions about my grading philosophy, please read the document I have posted at <http://www.cs.ucf.edu/~dmarino/ucf>. This document now has a complete history of grades I've given in every UCF class I've ever taught and the grade lines in every course I've taught here. **Note: This grading breakdown is subject to change. Any changes will be discussed in class.**

### **Individual Programming Assignments**

All programming assignments will be turned in over WebCourses. It's critically important to do these assignments in order to aid understanding of the course material. In order to grasp the material fully and feel comfortable programming in any language, one needs to write **MORE** programs than are assigned. Thus, while I'll have 10 graded programming assignments, it is expected that students will write or edit 2-3 programs a week **on top of** the assigned programming assignments, in order to get the proper practice. In addition, several practice exercises will be suggested on the course website. Students are encouraged to work on these exercises and get as much help on them as necessary from the course staff or peers to help them understand the course material. This practice is especially important for beginning students. If there are any ambiguities in any program description, please just ask me for my intent. **The due dates for each assignment will be posted on WebCourses ONLY.**

### *Late Assignment Policy*

**NO LATE HOMEWORK ASSIGNMENTS WILL BE ACCEPTED.** Due to possible server issues, it is **strongly suggested** that you attempt to submit programs **at least three hours** before the actual time it's due.

### **Exams**

For each exam, you may be allowed to use some notes as aids. The specifics for the allowable aids will be given in the class period before each quiz/exam.

### **Community Service Opportunity**

To get automatic full credit for 25 points (out of 150) on the Final Exam, you can do 5 hours of community service with a registered 501(c)(3) organization **BEFORE November 30, 2026**, and turn in the required signed form and activity summary (more details on the course web page) by the **12:00 pm on November 30, 2026**.

**You must submit the form and write up to me physically in my hand, either at the end of class or in my office BEFORE the stated deadline.**

Note, you will only get full credit if I receive the signed form and write up by 12:00 pm on that day. If I receive it at 12:10 pm on November 30<sup>th</sup>, then you'll have to take the appropriate portion of the Final Exam to earn points. Every semester, a couple students are just a couple minutes late and I don't count their forms. **Please do the community service early and submit the forms to me way in advance, so that this doesn't happen to you.**

### **AI Policy**

Try to limit your use of AI while in the beginning learning phases. My experience has been that the more students rely on AI, the less they actually learn. A number of studies are now coming out showing the negative effects of using AI for the purposes of education. Here is a link to the summary of one such study: <https://time.com/7295195/ai-chatgpt-google-learning-school/>.

The problem with using AI early in the learning process (other than the fact that you turn your brain off) is that AI still makes mistakes and it writes code that's very difficult to debug. But to get to the point where (a) you can realize the AI is wrong, and (b) you can fix it, you have to learn all the basics from the ground up and fully understand how things work. (This is similar to how mathematics students aren't allowed to use calculators until high school. Or at least, they shouldn't be allowed to use them until high school.)

### Course Assistance

You may attend the instructor's office hours or TA's office hours for general help understanding course material as well as help debugging individual programming assignments. It is strongly encouraged that you bring questions from the ungraded practice exercises to office hours to aid in understanding the material.

### Tentative Schedule

| Week            | Monday                              | Wednesday                              | Sunday         |
|-----------------|-------------------------------------|----------------------------------------|----------------|
| Aug 24 – 28     | Java Types, Vars, Arith Expr        | If                                     | <b>P1 Due</b>  |
| Aug 31 – Sept 4 | Loops                               | Math, Random, String                   | <b>P2 Due</b>  |
| Sept 7 – 11     | <b>Labor Day</b>                    | Static Methods                         |                |
| Sept 14 – 18    | Arrays                              | 2D Arrays                              | <b>P3 Due</b>  |
| Sept 21 – 25    | <b>Exam #1</b>                      | Intro to Classes                       |                |
| Sept 28 – Oct 2 | Instance Methods/Method Overloading | Class Design                           | <b>P4 Due</b>  |
| Oct 5 – 9       | Inheritance                         | Polymorphism                           | <b>P5 Due</b>  |
| Oct 12 – 16     | Interfaces/ Comparable Interface    | Abstract Classes                       | <b>P6 Due</b>  |
| Oct 19 – 23     | <b>Exam #2</b>                      | ArrayList/ArrayDeque                   |                |
| Oct 26 – 30     | HashSet/HashMap                     | TreeSet/TreeMap<br><b>WD Deadline</b>  | <b>P7 Due</b>  |
| Nov 2 – 6       | PriorityQueue                       | Exceptions                             | <b>P8 Due</b>  |
| Nov 9 – 13      | File I/O                            | <b>Veteran's Day</b>                   |                |
| Nov 16 – 20     | File I/O                            | Java Swing                             | <b>P9 Due</b>  |
| Nov 23 – 27     | <b>Thanksgiving</b>                 | <b>Thanksgiving</b>                    |                |
| Nov 30 – Dec 3  | Java GUI Example                    | FE Review                              | <b>P10 Due</b> |
| Dec 7 – 11      | <b>NO CLASS</b>                     | <b>Final Exam<br/>Dec 9 (10am-1pm)</b> |                |

**Note: Assignments will be given in class and will be due over WebCourses. Tentative dates are given above for the assignments but consult WebCourses for the final due dates and times. Also, this schedule may change based on the pace of lectures.**