COSI12b
Advanced Programming Techniques

Fall 2018
Mon, Wed, and Thu 11:00 AM-11:50 AM,
Room: Gerstenzang 122

Recitation: No recitation

Exams (all 6:30-7:20pm in Gerstenzang 121)
1. Thursday October 11
2. Thursday November 08
3. Thursday December 06

Instructor: Olga Papaemmanouil
E-Mail: olga@cs.brandeis.edu
Office: Volen 214
Office Hours: Wed, Thu, 12:00 PM-1:00 PM, or by appointment
Overview

The course will introduce students to object oriented programming using Java. It will focus on more sophisticated features such as design of classes, interfaces, packages, and APIs. It will also cover the basic principles of software design, testing and collaborative programming.

Upon completion of this class, students will be able to understand the concept of object-oriented programming (OOP) as well as the purpose and usage of inheritance, polymorphism, encapsulation and method overloading. They will be able to create Java application programs using sound OOP practices (e.g., interfaces and APIs) and proper program structuring. Finally, they will be able to develop programs using the Java Collection API as well as the Java standard class library.

Prerequisites

COSI 11a (or programming experience in C).

Experiential Learning

COSI 12B -1 (Advanced Programming Techniques) is an experiential learning course that aims to introduce the principles of object-oriented programming (OOP). Students will have the opportunity to implement weekly programming assignments (in Java), which simulate real-world applications, allowing them to build solid foundations in programming and software engineering. The first part of the class, introduces the basic concepts of OOP. It then offers hands-on experience on program design by introducing advanced OOP techniques such as interfaces, abstract classes and the Java collection framework. In the last part of the class, students learn (through in-classes implementation examples and assignments) how to leverage OOP to implement advanced data structures (e.g., linked lists, binary trees, etc) facilitating their transition to more advanced CS courses and topics. Students are also invited to participate into one-to-one meetings with TAs where their have the opportunity to demonstrate their assignments prior to grading and receive feedback.

Learning Objectives

The objective of the course is to:

- Cover issues related to the definition, creation and usage of classes, objects and methods.
- Discuss the principles of inheritance and polymorphism and demonstrate through problem analysis assignments how they relate to the design of methods, abstract classes and interfaces.
- Provide the foundation of good programming skills by discussing keys issues in the design of object-oriented software, including programming design patterns and programming testing.
- Cover the basics of creating APIs as well as allow students to explore the Java Abstract Programming Interface (API) and Java Collection Framework through programming assignments.
Textbook


✓ The textbook is optional, however, it makes a useful supplement to the lecture presentations. It contains practice problems and online videos you can use to study for your exams.

- Lecture notes and sample programs will be posted on LATTE.

Software

The recommended software for the course is the Java Development Kit (JDK) and the Eclipse editor.

Grading

The final grades for the course will be determined using the following weights:

- Homework: 35%
  ✓ Programming Assignments (roughly every week). Programs will be graded on “external correctness” (behavior) and “internal correctness” (style and design).
  ✓ Each student receives 5 "late days" for use on homework assignments. A late day allows you to submit a program up to 24 hours late without penalty. For example, you could use 2 late days and submit a program due Thursday 11pm on Saturday by 11pm with no penalty. Once a student has used up all the late days, each successive day that an assignment is late will result in a loss of 5% points on that assignment if submission is 1 day after the deadline and 10% points for every additional day. Regardless of how many late days you have, you may not submit a program more than 3 days after it is due.

- Exams: 55%
  ✓ Three exams: October 09, November 09, and December 07. Exams are scheduled during the recitation time.
  ✓ Make-up exams will not be given except in the case of a serious emergency.

- Participation: 10%

Success in this 4 credit hour course is based on the expectation that students will spend a minimum of 9 hours of study time per week in preparation for class (readings, papers, discussion sections, preparation for exams, etc.).

Class Rules

No laptops, tablets, or phones during lectures unless requested by the instructor

I find the students’ use of laptops, tablets, or phones in class is distracting to other students and to me. If you feel that your learning will be hampered by not having access to your laptop for note taking or other legitimate purposes, please speak to me. Otherwise, keep your devices turned off and stowed away during class.
Academic Honesty

As stated in the Rights and Responsibilities handbook, "Every member of the University community is expected to maintain the highest standards of academic honesty. A student shall not receive credit for work that is not the product of the student's own effort."

Programming assignments must be completed individually (unless specified otherwise by the instructor); all code you submit must be your own work. You may discuss general ideas of how to approach an assignment, but never specific details about the code to write. Any help you receive from or provide to classmates should be limited and should never involve details of how to code a solution.

As a student of this course you are agreeing to the following rules:

- You may not work as a partner with another student on an assignment.
- You may not get code from online sources.
- You may not show another student your solution to an assignment, nor look at his/her solution, for any reason.
- You may not have another person "walk you through" an assignment, describe in detail how to solve it, or sit with you as you write it. You also may not provide such help to another student. This includes current or former students, tutors, friends, TAs, web site forums, or anyone else.
- You may not post your homework solution code online or ask others for online help. This includes public message boards, forums, file sharing sites and services, or any other online system.
- Under our policy, a student who gives inappropriate help is equally guilty with one who receives it. Instead of providing such help to someone who does not understand an assignment, point him or her to other class resources such as lecture examples, the textbook, or emailing a TA or instructor. You must not share your solution and ideas with others. You must also ensure that your work is not copied by others, such as making sure to log out of shared computers, not leaving printouts of your code in public places, and not emailing your code to other students or posting it on the web. We enforce this policy by running similarity detection software over all submitted student programs.