

EECE 310: Software Engineering

Staff and Contact:

Instructor: Karthik Pattabiraman (karthikp@ece.ubc.ca) – All emails must have EECE 310 in subject

Teaching Assistants: Anna Thomas (annathomas87@gmail.com), Frolin Ocariza Jr. (focarizajr@yahoo.com)

Website: <http://courses.ece.ubc.ca/310/>

Course Logistics (Lectures, Office hours and Lab):

What ?	When ?	Where ?
Lectures	Tuesdays and Thursdays (2:00 to 3:30 PM)	MCLD 242
Karthik's Office hours	Tuesday (1:00 to 1:45 PM), other times by appointment	Fred Kaiser 4048
Programming Laboratory	As per your session	TBD

Course Introduction

This is a third-year course on software engineering. The goal of this course is to teach you to build and maintain large software systems. We will use Java as the main programming language, though this is NOT a course about Java. We will cover the following topics in class (rough outline):

- Modular decomposition
- Abstractions and Types
- Requirements Engineering and Specifications
- Testing and Iterative Development
- Verification and Validation
- Concurrent programming

Pre-requisites and background

The pre-requisite for this class is **CPSC 260: Object-Oriented Program Design**. **You are solely responsible for all material covered in the earlier class.** If you have not taken CPSC 260, then please reconsider your decision to take EECE 310 unless you have learnt the material by some other means (e.g., classes at other institutions).

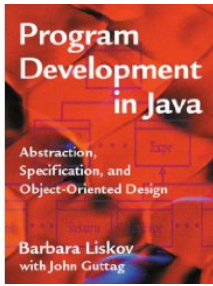
In addition, familiarity with the Java programming language is required to complete the assignments. If you know C++ well, then Java should be easy to pick up. **We will cover only the basics of the Java language in the lectures. Therefore, you will be solely responsible for learning enough Java in order to complete the assignments.**

Course Grading

We will have regular programming assignments in this class (total of 5 assignments over 12 weeks). Each assignment will count for 10% of your total grade and will cumulatively constitute 50 % of the course grade. Assignments will need to be done in teams of two – please choose your partner by the second week of class.

In addition to the assignments, we will have two exams – a mid-term exam and a final exam. **These will account for 15% and 30% of your grade respectively.** You need to earn a passing grade in the exams to pass the class. In other words, you will need to earn a minimum of 22.5 % of the course grade through the exams alone. **Otherwise, your final grade for the class will default to an F even if you cumulatively scored more than 50%.** Finally, you will get 5% of the marks for participating in class and in online discussions. We will also have a few practice quizzes in class, but these will not be graded. You are encouraged to use these quizzes for exam preparation.

Textbook and lecture Notes



The textbook for the class is “Program Development in Java” by Liskov and Guttag (Addison-Wesley ‘00). This book should be available from a variety of retailers including the UBC bookstore. However, we will not be covering all the chapters of the book in this class. You will not be tested on the chapters not covered in class. Lecture notes will be posted on the course web-page approximately every week. You will also be responsible for topics covered in class even if they are not mentioned in the textbook.

Finally, the programming assignments will require knowledge of Java beyond that covered in class. A useful reference for the Java language is the book “Thinking in Java”, third edition by Bruce Eckel. The book is available freely online.

Learning Objectives

1. Identify data-, procedural- and iteration- abstractions in software systems and use them appropriately
 - a. Write specifications for procedural abstractions
 - b. Understand exception handling mechanisms and how to use them
 - c. Define and use data abstractions and abstract data types
 - d. Implement iterator and generator abstractions
 - e. Understand type hierarchies and Liskov substitution principle
 - f. Define and implement polymorphic abstractions
2. Write specifications of software systems and ensure that the implementation meets the specification
3. Develop unit and integration tests using automated tools and rigorous techniques
4. Create data abstractions and demonstrate their correctness through formal techniques
5. Write correct, multi-threaded code without introducing deadlocks, race conditions.

Course Policies

1. This course has a zero tolerance policy towards plagiarism. Plagiarism constitutes any and all forms of sharing ideas, designs or implementation details. You can find more information about what constitutes plagiarism here: <http://www.library.ubc.ca/clc/airc.html>. The penalty for plagiarism can range from getting an F in the class to dismissal from the university: <http://www.students.ubc.ca/calendar/index.cfm?tree=3,54,111,960>
2. Assignments will be done in groups of two. You need to pick a partner by the beginning of the second week. You are free to discuss, share and collaborate with your partner without reservations. However, both of you will be jointly responsible for the solution you turn in. You may not share code or discuss ideas with any other pair of students working on the assignment. Further, both you and your partner will be awarded the same grade for the assignment. You are welcome to split up the work anyway you want. However, both of you will be expected to know the details of your solution to each assignment. We reserve the right to call upon you individually to explain the details of your solution. Failure to do so will result in you alone earning a 0 for the assignment.
3. The assignments cumulatively build upon each other, so not turning in even a single assignment can adversely impact your grade for future assignments. You will need to turn in the assignment in the lab on which it is due. If you cannot come to the lab to turn in the assignment, you must make arrangements with us ahead of time.
4. All deadlines are hard unless you have a documented medical or family emergency. In both cases, you may be called upon to produce documentation related to the nature of the emergency.
5. Makeup exams will be scheduled for those who miss exams only if a valid reason is provided (see UBC policies). If you have a conflict with the exam date/time, you must get my approval by email well in advance (minimum of one week’s notice), so that alternate arrangements can be made. If you miss an exam without getting my prior approval or having a documented emergency, you will receive a 0 for the exam.
6. We will use Piazza for online discussions and for answering questions about the assignments. You must sign up for a Piazza account by the end of this week. Questions about the lectures and the assignments must be posted to Piazza and NOT emailed to the TAs or the professor. You are also encouraged to answer others’ questions – this will earn you points for class participation.
7. Finally, it is your responsibility to keep up with course announcements, lectures and assignments by following the course webpage and Piazza. We will not use WebCT or Vista for posting announcements or assignments. **Failure to follow instructions will automatically earn you a 0 for the component being evaluated.**