COMPSCI 220 Programming Methodology

Syllabus

Description

Development of individual skills necessary for designing, implementing, testing and modifying larger programs, including: use of integrated design environments, design strategies and patterns, testing, working with large code bases and libraries, code refactoring, and use of debuggers and tools for version control. There will be significant programming and a mid-term and final examination. Prerequisite: CMPSCI 187 or ECE 242. 4 credits.

Course Materials

The required texts for this course is:

- Programming in Scala 2nd Edition, Martin Odersky, Lex Spoon, and Bill Venners. We will use this book as a reference for learning various aspects of the Scala programming language. http://www.amazon.com/Programming-Scala-Comprehensive-Step-Step/dp/0981531644

- Functional Programming in Scala, Paul Chiusano and Rúnar Bjarnason. We will follow Part 1 and Part 2 fairly closely. Although the book uses Scala as the language of choice the topics and concepts that it (and we) will cover are general functional programming concepts. http://www.amazon.com/Functional-Programming-Scala-Paul-Chiusano/dp/1617290653

We will also be using this book:

- Scala for the Impatient, Cay Horstmann. We will be only referencing the first 11 chapters from this book. Fortunately, the first 11 chapters are available for free from Typesafe. https://www.typesafe.com/resources/e-book/scala-for-the-impatient

All other reading and material will be drawn from freely available sources.

Course Objectives

There are a number of important objectives that this course aims to cover. The primary objective is to expose you to principles, concepts, techniques, libraries, languages, and tools that will elevate your knowledge and expertise in programming from that of a novice to an advanced programmer. Here is a short list of explicit objectives that we target:

- To learn and apply version control.
- To understand and use good programming principles.
- To learn and apply the Scala programming language.
- To learn additional language concepts in Java.
- To learn about different programming paradigms.
- To understand and apply functional programming patterns.
- To learn and apply test driven development.
- To learn and use build systems.
- To learn and apply proper documentation techniques of source code.
- To learn how to communicate to others through code and implementation.
- To understand larger code bases and navigate them effectively.
- To build libraries for others to use.
- To use and integrate external libraries.
- To learn, recognize, and apply software patterns.

- To learn about immutability and referential transparency.
- To learn and understand code reuse.
- To understand the difference between inheritance and composition.
- To recognize data patterns and interact with databases programmatically.
- To learn, apply, and implement regular expressions.
- To learn and apply custom data structures and algorithms.
- To use built-in language supported data structures and techniques.
- To understand, recognize, and implement domain specific languages.
- To learn and apply parsing techniques.
- To understand and apply interpretation and evaluation.
- To learn how to read and evaluate the code of others through peer review.
- To learn and use the command line.
- To learn and use integrated development environments.
- To learn how to refactor existing code.
- To learn the basics of software design.
- To learn how to evaluate basic software performance.

Course Assessment

Student knowledge of the course will be assessed from a variety of angles including:

- Project Assignments
- Exercise Assignments
- Three Unit Exams
- One Final Exam
- i-Clicker

Assessment and Grading

Each assessment component is worth a fixed number of points. At any point during the course you can easily calculate your current grade by the number of points you have achieved with respect to the total number of points you can attain in the course. You should read the course policies to understand lateness.

- **Project Assignments (800 points)**: There will be 8 project assignments assigned during the course. Each project assignment will take you 1 to 2 weeks to complete. Each project assignment is worth 100 points. **Late assignments will not be accepted**. Project assignments will be submitted as Scala projects using Moodle. The project assignments will be automatically graded using both *public* and *private* unit tests. You will have access to the public unit tests to better understand how we are testing your submission. The private tests will be additional tests only available to us.

- **Exercise Assignments (200 points)**: There will be 8 exercise assignments. Each exercise assignment will be released on the day of discussion and will be due at the beginning of the next day of discussion. Each exercise assignment is worth 25 points. Late submissions will not be accepted. Exercise assignments will be submitted as a Scala project using Moodle.

- **Unit Exams (300 points)**: There will be three unit exams. The unit exams will be taken online through Moodle and must be completed during the available time. Each unit exam will take approximately 1 hour to complete, however, you will have 2 hours. You will be able to take the online exams at any point within a three-day time window. Each unit exam is worth 100 points.

- **Final Exam (300 points)**: There will be 1 final exam. The final exam will be taken online through Moodle and must be completed during the available time. The final exam will take approximately 2 hours to complete, however, you will have 3 hours. The final exam will be available during the final exam period. The final exam is worth 300 points.

- **i-Clicker (80 points)**: We will routinely use i-Clicker in class for answering questions. You will not be graded on the correctness of your answer; however, we will record your answers as participation. Make sure to purchase an i-Clicker and register with this course on Moodle if you do not already have one.

The total number of points you can achieve in this course is **1680**. The approximate percentage of each assessment component with respect to your total grade is as follows:

- 48% Project Assignments (6% each)
- 12% Exercise Assignments (1% each)
- 18% Unit Exams (6% each)
- 18% Final Exam
- 5% i-Clicker

There is no opportunity for extra credit in this course. Unless otherwise specified, so do not ask! The typical breakdown of percentages and final grades for this course are A (93-100), A- (90-92), B+ (87-89), B (83-86), B- (80-82), C+ (77-79), C (73-76), C- (70-72), D+ (67-69), D (60-66), F (0-59). We do not scale grades.