Overview

Objective

The aim of CS111 is to introduce the student to the fundamental techniques used in computer science and software development. Upon completion of this course, the successful student should be able to design, write, test, and analyze programs to solve simple real-world problems.

CS111 uses the Java programming language, but it is not a training course in Java. Many Java features, such as applets, GUI programming, Java Server Pages (JSP), etc., are not covered in this course.

Prerequisite Knowledge

- Parts of a computer: processor, memory, disk drives, screen (monitor), keyboard, mouse
- File system concepts: files, directories (folders), difference between data files and programs
- Internet skills: use of web browser and e-mail
- Algebra concepts: variables, functions, exponentiation, logarithms

Fundamentals of Programming

Concepts

- The programming process
- Flowcharts
- Input, output, and temporary storage (variables)
- Mathematical computation
- Boolean logic
- Decision-making (branching)
- Iteration (looping)
- Test case construction
- Program tracing

Introductory Java Programming

Concepts

- Compile-time and run-time errors
- Objects and references
- Code instrumentation

Java

- Basic program framework and syntax rules
- Data types: boolean, int, double, char, objects/references
- Literals, variables, constants
- Operators, expressions, operator precedence
- Type conversion

- Instantiating objects, primitive and reference types
- null keyword
- Calling methods (static and non-static)
- Decision statements: if, if/else, if/else if/.../else
- Loops: do/while, while, for

Procedural Programming

Concepts

- Procedures (a.k.a. subroutines, functions, methods, etc.)
- Solving problems using recursion

Java

- Writing static methods
- Writing recursive methods

Data Structures

Concepts

- Strings
- Arrays (one-dimensional, multi-dimensional)

Java

- Strings
- Arrays (one-dimensional, two-dimensional)

Efficient Algorithms

Concepts

- Efficiency analysis, big-O notation
- Sequential search, binary search
- Sorting: selection sort, insertion sort, quicksort

Java

- Implementation of searching and sorting algorithms

Object-Oriented Programming

Concepts

- Object-oriented design, encapsulation
- Writing a test driver
- Blackbox testing

Java

- Defining classes, member variables, and methods
- public and private modifiers

Miscellaneous topics (time permitting)

Concepts

- Exceptions
- Files

Java

- try/catch statement
- throws statement
- File I/O

Research Statement

As Rutgers is a research university there is a possibility that by enrolling in this class you may be asked to participate in a research study. Participation in any such study will be optional and at no time will participation in a research study be part of a grade or a requirement for this course. This notification does not imply that by enrolling in this class you have provided consent to be a subject in a research study. Should you be asked to participate in a research study a consent form will be presented to you describing the study and asking for your signature. Participation in research is always voluntary and refusing to participate will have no adverse effects on your standing in the course. To learn more about research at Rutgers University and Human Subject research go to

http://orsp.rutgers.edu/index.php?q=content/institutional-review-board-irb.

Policies

CS 111 - Spring 2015

Policies

Academic Integrity

You are responsible for reading and understanding the DCS Academic Integrity Policy. Read the overview as well as the specific policies for exams.

Instances of cheating will be punished by a zero grade, a failing grade for the course, and/or referral to your dean, at the discretion of the course staff.

Lectures

Attendance is expected. Much of the course material is only discussed in lecture, not in the textbook. Some material will not even be in the lecture slides. If you miss a lecture, you are responsible for getting class notes from a classmate.

If the instructor is not present at the beginning of class, do not assume that the class has been cancelled until at least 15 minutes after its scheduled start.

Recitaton

If the Peer Leader is not present at the beginning of class, do not assume that the class has been cancelled until at least 15 minutes after its scheduled start.

Assignments

Graded exercises are assigned roughly weekly. These typically require writing or completing programs, and will often also involve written work. Assignments are to be submitted via Sakai.

Assignments will have a "due date" and a "grace period".

- If you have technical problems doing the assignment, e.g. your computer dies or the software does not seem to work for you, and it is before the time the assignment is due, you should contact a member of the course staff, who will work with you to try to resolve the problem.
- After the due time, but during the grace period, you can still turn in your assignment for full credit, and the 111 staff will still do their best to help you, but any problems you encounter will not constitute a valid excuse for failure to hand in the assignment on time.
- After the grace period, you may no longer turn in your assignment.
- Note that the deadline is based on the clock at the server running Sakai, which may not match your clock exactly. Also note that if Sakai is busy it may take some time before it gets to your request. If the deadline passes while you are waiting, your assignment will not be accepted.

  In other words, if you try to hand in your work close to the end of the grace period, you may be too late.

If you miss the deadline, do not e-mail your assignment to an instructor or TA. Such e-mails will be promptly deleted.

Your homework grade counts for only a small part of the course grade but doing the homework is essential to learning the material and doing well on the exams.

Programming assignments will generally be graded by an automatic script. Your program must compile without errors - otherwise you will not receive any credit for the assignment. For each problem, your code will be tested using a suite of test cases, and you will receive credit for each case on which your code performs correctly.

Do not ask for a regrade simply because you think you deserve more partial credit. If your program does not perform correctly on a particular test case, you will not receive points for that test case.

Exams

All exams are closed book and closed notes.

You will be required to show your student ID (or driver's license or other photo ID) at all exams. If your picture is rubbed out on your student ID, be sure to bring your driver's license.

If you cannot attend an exam, you must inform your instructor at least one full week in advance in order to be allowed to take a makeup exam (except in case of emergency). Makeups will be given only for religious obligation, verifiable medical issues, or other verifiable reasons. In emergency cases you must contact your instructor by email **as soon as possible**. For any reasons other than religious obligation, you must bring documentation explaining your need to take the makeup to your instructor. Appropriate types of documentation include:

- A doctor's note.
- An official police department accident report.
- etc.

Grading

Your grade will be computed as follows: We will record each grade you get as a percentage of the maximum possible score for that item. We will average your lab scores and (separately) your homework scores and then compute a weighted average with the following weights:

Type of score	Weight
Recitation Attendance	10%
Project	10%
Homework	10%
Codelab	10%
Midterm Exams (15% each)	30%
Final Exam	30%

The weighted average will be turned into a letter grade according to the following scale (TBD):

from	upto	grade
		A
		B+
		B
		C+
		C