Syllabus – CIS219 – Computer Programming 1 – Spring 2018

Instructor: Ken Mead
Email: kjmead@genesee.edu
Office location: D395 Math Science Area, Building D, Main Campus
Office hours: TR 9:30-10:10 and 12:10-12:45, W 2-4:30pm
Phone: 585-343-0055 x 6381
Homepage: http://faculty.genesee.edu/kjmead

IMPORTANT – Students enrolled in section CIS219-66: TESTS WILL BE PROCTORED. See the **Proctored Testing** section below.

Catalog description: Introduces computer concepts and programming in a modern, high-level language. Demonstrates computing system concepts, problem solving, and systematic program development in problems from a variety of application areas. Topics include problem analysis, algorithm design, top-down development, program testing and documentation, data types, input/output, sequence, selection, loops, data manipulation, functions, arrays, records, sets, strings, files, recursion, and an introduction to sorting, searching and other basic algorithms. Students should plan sufficient time to complete the necessary programming projects using the college's computing facilities.

Prerequisite: CIS125

Course Student Learning Outcomes (CSLOs):

The main objective of this course is for students to learn fundamental computer concepts and program development using a modern, high-level languages, such as Java. At the completion of this course, students will:

1. Correctly use the syntax and semantics of the language to create object oriented programs.

2. Write a one page summary documenting the 5 steps in the program development process as it applies to procedural-oriented programming.

3. Apply programming style and methodology, such as code format, modularity, commenting, documentation, structured design, pseudocoding and algorithm development, testing, debugging, and data validation in a minimum of 7-10 assignments requiring logical programming skills.

4. Develop a minimum of 5 programs which solve problems from a variety of areas using object oriented methods, creating applications that use objects. Language elements, such as, data types, I/O, sequence, selection, loops, data manipulation, member functions, arrays, records, sets, strings will be required.*

5. Demonstrate familiarity with the syntax of the language, logic patterns, and object oriented concepts such as encapsulation, inheritance as documented by multiple unit tests covering these terms/skills.

* This course objective has been identified as a student learning outcome that must be formally assessed as part of the Comprehensive Assessment Plan of the college.

Content Outline:

1. The programming environment
2. The program development process and good programming style
3. Variables, input/output, operators, comments
4. Decisions: if, if-else, else-if, switch, conditional operator
5. Loops: For, While, Do While
6. Nested Loops
7. Methods
8. Arrays, strings
9. Sort and search techniques
10. Introduction to OOP

Course Overview:

In this course, we'll be focusing on the Java programming language to learn how to program. Java is the most popular language in use on the planet (see the Tiobe Index at https://www.tiobe.com/tiobe-index/) , and is running on 3 billion + devices around the globe.

Here is a very general week-by-week description of the content of the course. This is by no means set in stone and should only be used as a guideline.

Week 1
Simple java programs to print "Hello World", and other fun stuff.
Overview of coding, compiling and executing a Java Program.
Programming design and structure.
Blocks of code and curly braces. Indenting.
Tools for writing Java Programs (textpad, BlueJ, NetBeans, etc).
Introduction to Linux and the use of putty.exe to connect to the csnlinux server.

Week 2
A very simple video introducing Java Variables
Order of precedence - Java math operators, integer Math, MathDemo.java
The Miles-per-Gallon Problem
More linux commands for file management, making folders, changing folders, pwd, ls, mkdir, rm, cd
The notion of CLASSPATH

Week 3
Reusable code, store more classes in the classpath
Declaring and initializing different data types within a program
Numeric Literals
Java Math operators (video) == ++ -- += /= %=
Escape Sequences \n \t
Basic Arithmetic Operators in Java Math.pow(base, Exponent) * / %
Advanced Arithmetical Operators

Week 4
Chapter 3
Java Comparative Operators Logic and Operators
if/else statements
Example with if's (practice)

Week 5
Continue with conditional logic
Start to use "static" Methods
More on Methods, Classes and Objects
Boolean Variables
Operator Precedence

Week 6
Review for test 1.
Test # 1

Week 7
Looping
Java operators such as pre-increment ++x, post-increment x++

Week 8
Nested Loops
Methods

Week 9
Methods
Method Overloading

Week 10
Test 2

Weeks 11 & 12
Java Arrays

Week 13 and 14
Finish Arrays
Introduction to OOP.

Weeks 14 & 15
Finish OOP
Reading and writing data from Disk
Review for Test

Week 16
Test # 3
Semester Wrap Up

Required materials:

Textbook: Introduction to Java Programming and Data Structures, Comprehensive Version (11th Edition), Daniel Liang, Pearson Publishing, ISBN: 978-0134670942

Proctored Testing: Students in section 01 of this course will take exams in class on the dates listed above. Missed exams will result in a grade of zero unless the student has an unavoidable reason for missing the test AND you notifies his/her instructor in advance (either in person, by sending email, or by phone call and leaving voicemail).

Students in section 66 should plan on taking their exams during the scheduled exam time and location as well. However, if this is not possible, the student needs to make arrangements to take each test either:

a) with the instructor at a different time, or,

b) at one of GCC's testing centers during their regularly scheduled testing hours. (see http://www.genesee.edu/offices/cap/testing/ for all the details on testing hours and locations).

In either case, the student needs to notify the instructor at least one week in advance with the game plan. Your instructor understands that the campus centers have very limited testing hours, and he will work with you to allow you to take your test within a reasonable, one to two day, window around the scheduled test date.

Grading:
Final grades are assigned according to the following scheme, with the final average rounded to the nearest integer (in %): 92 or higher = A, 90-91 = A-, 88-89 = B+, 82-87 = B, 80-81 = B-, 78-79 = C+, 72-77 = C, 70-71 = C-, 68-69 = D+, 62-67 = D, 60-61 = D-, 59 or less = F.

Grades will be weighted as follows:

- 60% - three in-class exams on 2/20, 4/3, and 5/8
- 40% - Programming projects and homework assignments.

To compute your final average, divide your total number of points by 5, round to the nearest integer, and compare to the scale above.

Breakdown of Projects and Assignments:

There will be many short programming assignments that must be completed throughout the semester. These will be graded as either complete or not, all or nothing.

There will also be between 3 and 5 larger programming projects which must be completed to specifications provided by the instructor.

Student Responsibilities:

- Come to class, especially on exam day! If you must miss an exam due to extenuating and unavoidable reasons, you need to contact your instructor beforehand to re-schedule. Missing an exam without permission and good reason will result in a grade of zero.

- Complete projects on time! Late penalties will apply as follows: 10% within 3 days of deadline, 20% within first week of deadline, 40% after the first week. **If a project is not completed at all, it will receive a grade of zero and 2 points will be deducted from your overall average.**

- Do your own work! This means no cutting and pasting code from the internet. Students are encouraged to discuss programming assignments and projects, and even develop overall strategies for solving problems together, but must ultimately write their own code. Any evidence of plagiarism will result in a grade of zero for the assignment or project. Multiple offenses will result in a grade of F for the course.

- Stay engaged! Come up with ideas for problems to solve! Enjoy the material because it's really good stuff!!!

- Put the phone away. Respect your classmates. Don't be disruptive. Blah blah blah. You know the drill.

The instructor reserves the right to make any reasonable and necessary modifications to the statements above. This document is subject to change.