IMPERIAL VALLEY COLLEGE

Basic Course Information

Semester:	**Sprint 2022**	Instructor Name:	**Octavio Ortiz**
Course Title & #:	**CS 221**	Email:	**octavio.ortiz@imperial.edu**
CRN #:	**20549**	Webpage (optional):	**Canvas**
Classroom:	**801**	Office #:	**2767.1**
Class Dates:	**2/14/22 – 6/10/22**	Office Hours:	**M: 9:45 – 10:15 AM** **T: 5:30 – 6 PM** **T/TH: 9 – 10 AM** **TH: 4:40 – 5:10 PM (IHS)**
Class Days:	**Monday/Wednesday/Friday**	Office Phone #:	**760-355-5706**
Class Times:	**8 – 9:25 AM, 8 – 10:05 AM**	Emergency Contact:	**Silvia Murray: 760-355-6201**
Units:	**3**	Class Format:	**Face-to-Face + Online/Zoom**

Course Description

Introduction to programming and software engineering for computer science majors and computer professionals. A systematic approach to the design, implementation, and management of robust Java computer programs. Course emphasizes Object Oriented programming design, programming documentation, testing and debugging techniques. (C-ID COMP 122) (CSU/UC)

Course Prerequisite(s) and/or Corequisite(s)

None

Student Learning Outcomes

Upon course completion, the successful student will have acquired new skills, knowledge, and or attitudes as demonstrated by being able to:

1. Correctly use classes from the standard Java libraries to solve a problem
2. Correctly use graphical user interface (GUI) components to create a program
3. Correctly use inheritance relations to solve a problem

Course Objectives

Upon satisfactory completion of the course, students will be able to:

1. Analyze unstructured problems and design computer solutions
2. Use procedural techniques to control program flow (sequence, selection and repetition) and declare local variables and pass parameters to functions.
3. Demonstrate object-oriented programming language syntax and structure
4. Define and use classes and methods to implement algorithms
5. Assess the applicability of common algorithms to specific program design problems
6. Develop and use beginning program testing data and techniques
7. Assess the applicability of common data structures to specific program design problems
8. Use system debuggers
9. Adhere to style and documentation standards in writing programs

IMPERIAL VALLEY COLLEGE

Textbooks & Other Resources or Links

Introduction to JAVA – Programming and Data Structures
Author: Y. Daniel Liang
Edition: 11th
ISBN: 978-0-13-467094-2
Copyright Year: 2018
Publisher: Pearson Prentice Hall

Course Requirements and Instructional Methods

Students will be exposed to various instructional methods. Lectures, both in person and through pre-recorded tutorial videos, will introduce students to fundamental programming concepts. Students will then apply what they learn in lectures to their own programming assignments and applications. Guidance and modeling will be provided during the face-to-face component of the course.

Programming assignments will be relatively short and will assess a student's mastery of a particular programming skill, as well as a student's ability to problem solve. Programming applications, or projects, will be more intricate. To develop an application, students will rely on the various programming and problem-solving skills they have developed up to that point.

There will be short quizzes where students will read code and answer multiple choice, true-false, and free-response questions pertaining to the code segments. A comprehensive semester final exam will assess students' ability to read, debug and rationalize code segments that range in complexity.

Course Grading Based on Course Objectives

ASSIGNMENT	POINTS
Collaborative Notes	**25%**
Approximately 2-3 per week	
Programming Assignments	**40%**
Approximately 10-12 PA's in semester	
Quizzes/Discussions	**15%**
10 or fewer quizzes in semester	
Projects/Final Exam	**20%**
Midterm project & comprehensive final	
Total	**100%**

Score	Letter Grade
≥ 90%	**A**
≥ 80%	**B**
≥ 70%	**C**
≥ 60%	**D**
< 60%	**F**

IMPERIAL VALLEY COLLEGE

Course Policies

Attendance:

Students are expected to attend every class meeting. Lectures will preview programming assignments, programming applications and future assessments.

- Although attendance is not explicitly factored into your grade, failing to complete programming assignments and assessments due to absences will negatively impact your grade.

Late Submissions:

Programming assignments are to be completed and submitted by the due date stated on Canvas. Late programming assignments will be accepted and penalized as follows:

- 90% maximum score if submitted within 24 hours past due date
- 80% maximum score if submitted within 48 hours past due date
- 70% maximum score if submitted within 72 hours past due date
- 50% maximum score if more than three days and less than a week past due date
- No credit will be given to assignments that are one week or more past due

Programming applications/projects, quizzes and the final exam will NOT be accepted late.

Make-up Assignments:

There are no make-up assignments.

- Programming assignments that are more than a week past due will receive a score of 0 and cannot be made up.
- Programming applications/projects and quizzes cannot be made up, however, if the material is presented again in future applications or quizzes, then the failed assessment will be reevaluated.

Drop Policy

The instructor reserves the right to drop students who fail to attend the first-class session or fail to complete the first assignment by the assigned due date.

Other Course Information

Resources:

https://www.w3schools.com – Learn Programming

https://docs.oracle.com/en/java/index.html - Java Documentation

IVC Student Resources

IVC wants you to be successful in all aspects of your education. For help, resources, services, and an explanation of policies, visit http://www.imperial.edu/studentresources or click the heart icon in Canvas.

IMPERIAL VALLEY COLLEGE

Course Calendar

The semester calendar is meant to provide an overview of the topics that will be covered throughout the semester. Every effort will be made to adhere to the calendar; however, changes might be necessary.

Week	Date	Topic	Assignment
Week 1	2/14	• **Syllabus & Course Policies** ○ Modules, collaborative notes, programming assignments, etc…	
	2/16	• **Fundamentals** ○ Basic file structure in Java ○ Printing	
Week 2	2/21	• **Elementary Programming** ○ Data Types, String Objects	
	2/23	• **Elementary Programming** ○ String Objects, User Input	
Week 3	2/28	• **Selection** ○ `if`-statements, `AND/OR`, `NOT`, `MOD` operators	
	3/2	• **Selection** ○ `if-else`, `if-else if` statements	
Week 4	3/7	• **Strings & Mathematical Functions** ○ Common math functions	
	3/9	• **Strings & Mathematical Functions** ○ `String` type and its methods	
Week 5	3/14	• **Repetition** ○ `while` loops, `do-while` loops, `for` loops	
	3/16	• **Repetition** ○ Implementing Loops	
Week 6	3/21	• **Repetition & Arrays** ○ 1-D arrays, `for` loops & arrays	
	3/23	• **Repetition & Arrays** ○ `for-each` loops & arrays	
Week 7	3/28	• **Encapsulation** ○ Functions & Methods, Parameters, Return type	
	3/30	• **Encapsulation** ○ Implementing Methods	
Week 8	4/4	**Midterm Project**	
	4/6		
Week 9	4/11	• **Multidimensional Arrays** ○ 2D array, nested `for` loops & 2-D arrays	
	4/13	• **Multidimensional Arrays** ○ Implementing 2D arrays	
Spring Break			
Week 10	4/25	• **ArrayLists** ○ `add(), remove(), get(), isEmpty()`…	

Week	Date	Topic	Assignment
	4/27	• **ArrayLists** ○ Implementing arraylists	
Week 11	5/2	• **Encapsulation & Class Design** ○ Fields, constructors, the `this` reference	
	5/4	• **Encapsulation & Class Design** ○ Accessor & Mutator Methods, `toString()` method	
Week 12	5/9	• **Class Design** ○ Class Instantiation, zero & multiple argument Constructor, overloading methods	
	5/11	• **Class Design** ○ Static vs. non-static methods & fields	
Week 13	5/16	• **Class Design** ○ Passing Object to Methods	
	5/18	• **Class Design** ○ Array of Objects	
Week 14	5/23	• **Class Design** ○ Writing your own classes	
	5/25	• **Inheritance** ○ `extends` keyword, superclasses & subclasses, `super` keyword, overriding and overloading methods	
Week 15	5/30	• **Inheritance** ○ Advantages of inheritance	
	6/1	• **GUI** ○ JavaFX vs. Swing and AWT, JavaFX basic structure, Color class, Font class, Panes & Groups	
Week 16	6/6	• **GUI** ○ Implementing GUI's with JavaFX	
	6/8	• **Comprehensive Final Exam**	

*****Subject to change without prior notice*****