



IMPERIAL VALLEY COLLEGE

Basic Course Information

Semester:	Fall 2021	Instructor Name:	Octavio Ortiz
Course Title & #:	CS 221	Email:	octavio.ortiz@imperial.edu
CRN #:	10951	Webpage (optional):	Canvas
Classroom:	N/A	Office #:	2767.1
Class Dates:	8/16/21 – 12/11/21	Office Hours:	MW: 9:40 – 10:10 AM 6:00 – 6:30 PM T/TR: 9:10 – 10:10 AM
Class Days:	N/A	Office Phone #:	760-355-5706
Class Times:	N/A	Emergency Contact:	Silvia Murray: 760-355-6201
Units:	3	Class Format:	Online

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

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.

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 weekly discussion posts where students will reflect on their learning and engage in discussions regarding previous, current and new programming topics. A comprehensive semester final exam or project will assess students’ ability to read, debug and rationalize code segments that range in complexity.

Course Grading Based on Course Objectives

ASSIGNMENT	POINTS
Programming Assignments	40%
Approximately 10-12 PA’s in semester	
Programming Applications/Projects	30%
Three applications in semester	
Discussion Posts	10%
Routinely contribute to discussion posts	
Final Exam	20%
Comprehensive final due 12/10/21	
Total	100%

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

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 and discussion posts are to be completed and submitted by the due date stated on Canvas. Late programming assignments and discussion posts 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 the final exam/project 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 cannot be made up, however, if the material is presented again in future applications, then the failed assessment will be reevaluated.

Drop Policy

The instructor reserves the right to drop students who 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.

Course Calendar

Week	Date	Topic	Assignment
Week 1	8/16	<ul style="list-style-type: none"> • Fundamentals <ul style="list-style-type: none"> ○ Printing, Data Types, User Input ○ Arithmetic, Modulus, <code>Math.random()</code> 	
Week 2	8/23	<ul style="list-style-type: none"> • Selection <ul style="list-style-type: none"> ○ <code>if</code>-statements, AND/OR, NOT, MOD operators ○ <code>if-else</code>, <code>if-else if</code> statements 	
Week 3	8/30	<ul style="list-style-type: none"> • Strings & Mathematical Functions <ul style="list-style-type: none"> ○ Common math functions ○ <code>String</code> type and its methods 	
Week 4	9/6	<ul style="list-style-type: none"> • Repetition <ul style="list-style-type: none"> ○ <code>while</code> loops, <code>do-while</code> loops, <code>for</code> loops ○ Implementing Loops 	
Week 5	9/13	<ul style="list-style-type: none"> • Repetition & Arrays <ul style="list-style-type: none"> ○ 1-D arrays, <code>for</code> loops & arrays ○ <code>for-each</code> loops & arrays 	
Week 6	9/20	<ul style="list-style-type: none"> • Programming Application (Project 1) 	
Week 7	9/27	<ul style="list-style-type: none"> • Encapsulation <ul style="list-style-type: none"> ○ Functions & Methods, Parameters, Return type ○ Implementing Methods 	
Week 8	10/4	<ul style="list-style-type: none"> • Multidimensional Arrays <ul style="list-style-type: none"> ○ 2D array, nested <code>for</code> loops & 2-D arrays ○ Implementing 2D arrays 	
Week 9	10/11	<ul style="list-style-type: none"> • ArrayLists <ul style="list-style-type: none"> ○ <code>add()</code>, <code>remove()</code>, <code>get()</code>, <code>isEmpty()</code>... ○ Implementing arraylists 	
Week 10	10/18	<ul style="list-style-type: none"> • Programming Application (Project 2) 	
Week 11	10/25	<ul style="list-style-type: none"> • Encapsulation & Class Design <ul style="list-style-type: none"> ○ Fields, constructors, the <code>this</code> reference ○ Accessor & Mutator Methods, <code>toString()</code> method 	
Week 12	11/1	<ul style="list-style-type: none"> • Class Design <ul style="list-style-type: none"> ○ Class Instantiation, zero & multiple argument Constructor, overloading methods ○ Static vs. non-static methods & fields 	
Week 13	11/8	<ul style="list-style-type: none"> • Class Design <ul style="list-style-type: none"> ○ Passing Object to Methods ○ Array of Objects 	

Week	Date	Topic	Assignment
Week 14	11/15	<ul style="list-style-type: none"> • Programming Application (Project 3) 	
Thanksgiving Break			
Week 15	11/29	<ul style="list-style-type: none"> • Inheritance <ul style="list-style-type: none"> ○ extends keyword, superclasses & subclasses, super keyword, overriding and overloading methods • Interfaces <ul style="list-style-type: none"> ○ Abstract classes, implements keyword, overriding abstract methods 	
Week 16	12/6	<ul style="list-style-type: none"> • GUI <ul style="list-style-type: none"> ○ JavaFX vs. Swing and AWT, JavaFX basic structure, Color class, Font class, Panes & Groups • Comprehensive Final Exam/Project (Due 12/10/21) 	

Subject to change without prior notice