



Basic Course Information

Semester:	Spring 2026	Instructor Name:	Octavio Ortiz
Course Title & #:	CS 221	Email:	octavio.ortiz@imperial.edu
CRN #:	20549	Webpage (optional):	Canvas Course
Classroom:	4300	Office #:	2766
Class Dates:	2/16 – 6/10	Office Hours:	Faculty Schedule
Class Days:	M/W	Office Phone #:	760-355-5706
Class Times:	1:00 – 3:30 PM	Emergency Contact:	Silvia Murray
Units:	3	Class Format:	In-Person

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. Write an application that incorporates the following programming paradigms to solve a real-world problem: sequence, selection, repetition, functions (user-defined and built-in), variables (primitive and reference), user-input, arrays, and data structures.
2. Write an application that consists of various classes in an effort to manage the complexity of the application. Where one of the classes is written to serve as a blueprint from which objects can be instantiated.
3. Write an application that is interactive for the user, and incorporates Graphical User Interfaces (GUI's), with the use of JavaFX API's.

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
10. Implement JavaFX API's to develop applications that incorporate Graphical User Interfaces (GUI's)

Textbooks & Other Resources or Links

Introduction to JAVA – Programming and Data Structures

Author: Y. Daniel Liang
Edition: 13th
ISBN: 978-0138123536
Copyright Year: 2024
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.

In midterm exams 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
Programming Assignments	10%
Approximately 10-15 PA's	
Exams	60%
Three planned Exams	
Projects/Final Exam	30%
Midterm/Final project & comprehensive final	
Total	100%

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

Course Policies

Attendance:

Attendance is mandatory. 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.
- Students with excessive absences will be dropped from the course as outlined in AP 5075.

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 submitted past the hard deadline (see calendar)

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

Make-up Assignments:

There are no make-up assignments.

- 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.

Anticipated Class Schedule/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	2/16	President's Day	
	2/18	<ul style="list-style-type: none"> • Syllabus & Course Policies <ul style="list-style-type: none"> ○ Modules, programming assignments, etc... • Fundamentals <ul style="list-style-type: none"> ○ Basic file structure in Java ○ Printing, Data types 	
Week 2	2/23	<ul style="list-style-type: none"> • Elementary Programming <ul style="list-style-type: none"> ○ Data Types, String Objects ○ String Objects, User Input 	
	2/25	<ul style="list-style-type: none"> • Selection <ul style="list-style-type: none"> ○ If-statements 	
Week 3	3/2	<ul style="list-style-type: none"> • Selection <ul style="list-style-type: none"> ○ AND/OR, NOT, MOD operators 	
	3/4	<ul style="list-style-type: none"> • Strings & Mathematical Functions <ul style="list-style-type: none"> ○ Common math functions 	
Week 4	3/9	<ul style="list-style-type: none"> • Strings & Mathematical Functions <ul style="list-style-type: none"> ○ String type and its methods 	
	3/11	<ul style="list-style-type: none"> • Chapter 1-4 Review • Exam 1 (Chapters 1-4) <ul style="list-style-type: none"> ○ Deadline to submit late assignments (Ch.1-4) for 50% credit. 	
Week 5	3/16	<ul style="list-style-type: none"> • Repetition <ul style="list-style-type: none"> ○ while loops, do-while loops, for loops 	
	3/18	<ul style="list-style-type: none"> • Repetition <ul style="list-style-type: none"> ○ Implementing Loops 	
Week 6	3/23	<ul style="list-style-type: none"> • Repetition & Arrays <ul style="list-style-type: none"> ○ 1-D arrays, for loops & arrays 	
	3/25	<ul style="list-style-type: none"> • Repetition & Arrays <ul style="list-style-type: none"> ○ for-each loops & arrays 	
Week 7	3/30	<ul style="list-style-type: none"> • Encapsulation <ul style="list-style-type: none"> ○ Functions & Methods, Parameters, Return type 	
	4/1	<ul style="list-style-type: none"> • Encapsulation <ul style="list-style-type: none"> ○ Implementing Methods 	
Week 8	4/13	Midterm Project	
	4/15	<ul style="list-style-type: none"> • Exam 2 (Chapters 5-7) <ul style="list-style-type: none"> ○ Deadline to submit late assignments (Ch.5-7) for 50% credit. 	
Week 9	4/20	<ul style="list-style-type: none"> • Multidimensional Arrays <ul style="list-style-type: none"> ○ 2D array, nested for loops & 2-D arrays 	

Week	Date	Topic	Assignment
	4/22	<ul style="list-style-type: none"> • Multidimensional Arrays <ul style="list-style-type: none"> ○ Implementing 2D arrays 	
Week 10	4/27	<ul style="list-style-type: none"> • ArrayLists <ul style="list-style-type: none"> ○ add(), remove(), get(), isEmpty()... 	
	4/29	<ul style="list-style-type: none"> • ArrayLists <ul style="list-style-type: none"> ○ Implementing arraylists 	
Week 11	5/4	<ul style="list-style-type: none"> • Encapsulation & Class Design <ul style="list-style-type: none"> ○ Fields, constructors, the this reference 	
	5/6	<ul style="list-style-type: none"> • Encapsulation & Class Design <ul style="list-style-type: none"> ○ Accessor & Mutator Methods, toString() method 	
Week 12	5/11	<ul style="list-style-type: none"> • Class Design <ul style="list-style-type: none"> ○ Class Instantiation, zero & multiple argument Constructor, overloading methods 	
	5/13	<ul style="list-style-type: none"> • Class Design <ul style="list-style-type: none"> ○ Static vs. non-static methods & fields • Exam 3 (Chapters 8-10) <ul style="list-style-type: none"> ○ Deadline to submit late assignments (Ch.8-10) for 50% credit. 	
Week 13	5/18	<ul style="list-style-type: none"> • Class Design <ul style="list-style-type: none"> ○ Passing Object to Methods 	
	5/20	<ul style="list-style-type: none"> • Class Design <ul style="list-style-type: none"> ○ Array of Objects ○ Writing your own classes • Inheritance <ul style="list-style-type: none"> ○ extends keyword, superclasses & subclasses, super keyword, overriding and overloading methods 	
Week 14	5/25	Memorial Day	
	5/27	<ul style="list-style-type: none"> • Inheritance <ul style="list-style-type: none"> ○ Advantages of inheritance ○ Inheritance and JavaFX 	
Week 15	6/1	<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 	
	6/3	<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 • Final Project 	
Week 16	6/8	<ul style="list-style-type: none"> • GUI <ul style="list-style-type: none"> ○ Implementing GUI's with JavaFX 	
	6/10	<ul style="list-style-type: none"> • Final Project Due • Comprehensive Final Exam 	

Subject to change without prior notice



IMPERIAL VALLEY COLLEGE

OCTAVIO ORTIZ
IVC Instructor's Schedule of Classes and Office Hours
Science, Math and Engineering Department
Spring 2026

		7:00 AM	8:00 AM	9:00 AM	10:00 AM	11:00 AM	12:00 PM	1:00 PM	2:00 PM	3:00 PM	4:00 PM	5:00 PM
M	COURSE			Office Hours		ENGR 100		CS 221				
	CRN			8:40 - 10:10 AM		LEC/LAB 20981		LEC/LAB 20549				
	LEC/LAB			Online		10:15 - 12:45 PM		1:00 - 3:30 PM				
	FACULTY					RM #4300		RM #4300				
T	COURSE		ENGR 212			Office Hours		CS 231				
	CRN		LEC 20607			11:15 - 11:45 AM		LAB/LEC 20550				
	LEC/LAB		8:00 - 11:10 AM					1:00 - 2:25 PM				
	FACULTY		RM #2722			RM #2766		RM #4300				
W	COURSE			Office Hours		ENGR 100		CS 221				
	CRN			8:40 - 10:10 AM		LEC/LAB 20981		LEC/LAB 20549				
	LEC/LAB			RM #2766		10:15 - 12:45 PM		1:00 - 3:30 PM				
	FACULTY					RM #4300		RM #4300				
R	COURSE		ENGR 240			Office Hours		CS 231				
	CRN		LEC 21274			11:15 - 11:45 AM		LAB/LEC 20550				
	LEC/LAB		8:00 - 11:10 AM					1:00 - 2:25 PM				
	FACULTY		RM #2722			Online		RM #4300				
F	COURSE											
	CRN											
	LEC/LAB											
	FACULTY											

Course No	Sections	Lec Hrs	Lab Hrs
CS 221	1	2	3
CS 231	1	2	3
ENGR 100	1	2	3
ENGR 212	1	3	
ENGR 240	1	3	
Total		12	9

Instructor's Name	Octavio Ortiz
Phone	760-355-5706
Office No.	2766
Email	octavio.ortiz@imperial.edu