



Basic Course Information

Semester:	Fall 2022	Instructor Name:	Octavio Ortiz
Course Title & #:	CS 221	Email:	octavio.ortiz@imperial.edu
CRN #:	10962	Webpage (optional):	Canvas Course
Classroom:	3109	Office #:	2767.1
Class Dates:	8/15 – 12/10	Office Hours:	Faculty Schedule
Class Days:	T/TR	Office Phone #:	760-355-5706
Class Times:	4:20 – 5:45 PM	Emergency Contact:	Silvia Murray
Units:	3	Class Format:	Hybrid (In-Person/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: 12th

ISBN: 978-0-13-652023-8

Copyright Year: 2020

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.

Collaborative notes 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 assignments, and 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 final project and 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	10%
Approximately 1-3 per week	
Programming Assignments	15%
Approximately 5-10 PA’s in semester	
Quizzes	45%
3 Planned Quizzes	
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	8/16	<ul style="list-style-type: none"> • Syllabus & Course Policies <ul style="list-style-type: none"> ○ Modules, collaborative notes, programming assignments, etc... 	
	8/18	<ul style="list-style-type: none"> • Fundamentals <ul style="list-style-type: none"> ○ Basic file structure in Java ○ Printing 	
Week 2	8/23	<ul style="list-style-type: none"> • Elementary Programming <ul style="list-style-type: none"> ○ Data Types, String Objects 	
	8/24	<ul style="list-style-type: none"> • Elementary Programming <ul style="list-style-type: none"> ○ String Objects, User Input 	
Week 3	8/30	<ul style="list-style-type: none"> • Selection <ul style="list-style-type: none"> ○ if-statements, AND/OR, NOT, MOD operators 	
	9/1	<ul style="list-style-type: none"> • Selection <ul style="list-style-type: none"> ○ if-else, if-else if statements 	
Week 4	9/6	<ul style="list-style-type: none"> • Strings & Mathematical Functions <ul style="list-style-type: none"> ○ Common math functions 	
	9/8	<ul style="list-style-type: none"> • Strings & Mathematical Functions <ul style="list-style-type: none"> ○ String type and its methods 	
Week 5	9/13	<ul style="list-style-type: none"> • Quiz 1 (Chapters 1-4) <ul style="list-style-type: none"> ○ Deadline to submit late assignments (Ch.1-4) for 50% credit. • Repetition <ul style="list-style-type: none"> ○ while loops, do-while loops, for loops 	
	9/15	<ul style="list-style-type: none"> • Repetition <ul style="list-style-type: none"> ○ Implementing Loops 	
Week 6	9/20	<ul style="list-style-type: none"> • Repetition & Arrays <ul style="list-style-type: none"> ○ 1-D arrays, for loops & arrays 	
	9/22	<ul style="list-style-type: none"> • Repetition & Arrays <ul style="list-style-type: none"> ○ for-each loops & arrays 	
Week 7	9/27	<ul style="list-style-type: none"> • Encapsulation <ul style="list-style-type: none"> ○ Functions & Methods, Parameters, Return type 	
	9/29	<ul style="list-style-type: none"> • Encapsulation <ul style="list-style-type: none"> ○ Implementing Methods 	
Week 8	10/4	Midterm Project	
	10/6		
Week 9	10/11	<ul style="list-style-type: none"> • Quiz 2 (Chapters 5-7) <ul style="list-style-type: none"> ○ Deadline to submit late assignments (Ch.5-7) for 50% credit. 	

Week	Date	Topic	Assignment
		<ul style="list-style-type: none"> • Multidimensional Arrays <ul style="list-style-type: none"> ○ 2D array, nested for loops & 2-D arrays 	
	10/13	<ul style="list-style-type: none"> • Multidimensional Arrays <ul style="list-style-type: none"> ○ Implementing 2D arrays 	
Week 10	10/18	<ul style="list-style-type: none"> • ArrayLists <ul style="list-style-type: none"> ○ add(), remove(), get(), isEmpty()... 	
	10/20	<ul style="list-style-type: none"> • ArrayLists <ul style="list-style-type: none"> ○ Implementing arraylists 	
Week 11	10/25	<ul style="list-style-type: none"> • Encapsulation & Class Design <ul style="list-style-type: none"> ○ Fields, constructors, the this reference 	
	10/27	<ul style="list-style-type: none"> • Encapsulation & Class Design <ul style="list-style-type: none"> ○ Accessor & Mutator Methods, toString() 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 	
	11/3	<ul style="list-style-type: none"> • Class Design <ul style="list-style-type: none"> ○ 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 	
	11/10	<ul style="list-style-type: none"> • Class Design <ul style="list-style-type: none"> ○ Array of Objects ○ Writing your own classes • Quiz 3 (Chapters 8-10) <ul style="list-style-type: none"> ○ Deadline to submit late assignments (Ch.8-10) for 50% credit. 	
Week 14	11/15	<ul style="list-style-type: none"> • Inheritance <ul style="list-style-type: none"> ○ extends keyword, superclasses & subclasses, super keyword, overriding and overloading methods 	
	11/17	<ul style="list-style-type: none"> • Inheritance <ul style="list-style-type: none"> ○ Advantages of inheritance 	
THANKSGIVING BREAK			
Week 15	11/29	<ul style="list-style-type: none"> • Inheritance <ul style="list-style-type: none"> ○ Inheritance and JavaFX 	
	12/1	<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 • Final Project 	
Week 16	12/6	<ul style="list-style-type: none"> • GUI <ul style="list-style-type: none"> ○ Implementing GUI's with JavaFX • Final Project 	
	12/8	<ul style="list-style-type: none"> • Final Project Due • Comprehensive Final Exam 	

Subject to change without prior notice