IMPERIAL VALLEY COLLEGE

Basic Course Information

Semester:	**Spring 2021**	Instructor Name:	**Alanna Jennings**
Course Title & #:	**CS 221 – Intro to OOP in Java**	Email:	**Alanna.Jennings@imperial.edu**
CRN #:	**21125**	Webpage (optional):	**Canvas Course CS 221**
Classroom:	**Online**	Office #:	**Zoom link in Canvas**
Class Dates:	**2/16/21 – 6/11/21**	Office Hours:	**Wednesday 12-1 pm or by appointment**
Class Days:	**Monday/Tuesday**	Office Phone #:	**Email**
Class Times:	**(Mon 12-1pm) & (Tue 10-11am) optional zoom class times**	Emergency Contact:	**760-355-6201 (Silvia Murray)**
Lab Days:	**Online**	Class Format:	**Online**
Lab Times:	**Online**	Units:	**3.0**

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.

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

none

Student Learning Outcomes

Upon course completion, the student will have acquired the skills necessary to:

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:

- Analyze unstructured problems and design computer solutions
- Use procedural techniques to control program flow (sequence, selection and repetition) and declare local variables and pass parameters to functions.
- Demonstrate object-oriented programming language syntax and structure
- Define and use classes and methods to implement algorithms
- Assess the applicability of common algorithms to specific program design problems
- Develop and use beginning program testing data and techniques
- Assess the applicability of common data structures to specific program design problems
- Use system debuggers
- 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

How will the class be structured in the online modality?
ONLINE COURSE STRUCTURE

1. Zoom Lecture and Labs
   (Attendance highly suggested, but optional); will poll students to determine best time to meet most students.)
2. Video Lectures and Labs
3. Projects/Discussion Boards
4. Online Midterm and Final

How to Succeed in the ONLINE course structure:

• It will be imperative you keep up with the course and stay disciplined.

• Dedicate a time each day to watch videos and do homework. It is best if you break it up into multiple small intervals. This gives your brain some rest time.

• Attend virtual “zoom” TBD

Course Requirements and Instructional Methods

Each session will consist of a combination of lectures, group discussions, problem solving and reflecting on the concepts covered. Students will be encouraged to share their ideas with each other and with the class to promote active engagement. Programming assignments will be assigned weekly and checked for completion. Students will work on multiple projects that will assess their conceptual understanding of key topics in Java.

Out of Class Assignments: The Department of Education policy states that one (1) credit hour is the amount of student work that reasonably approximates not less than one hour of class time and two (2) hours of out-of-class time per week over the span of a semester. WASC has adopted a similar requirement.

Course Grading Based on Course Objectives

The semester will consist of weekly programming assignments that reinforce fundamental concepts in Java. These programming assignments are essential to the development of coding skills and a complete understanding of object-oriented programming languages. Thus, the programming assignments account for 55% of the overall grade. In addition to weekly programming assignments, students will work on programming applications at the culmination of key major topics in Java. These programming applications (projects) reinforce major topics in Java and tie in all the fundamental concepts implemented in the weekly programming assignments. The programming applications will account for 15% of the overall grade. Finally, at the culmination of the semester, a midterm and final exam will be administered which will account for 15% each for of the overall grade.

CATEGORY	PERCENT OF GRADE
Programming Assignments	55%
Projects	15%
Midterm	15%
Final	15%

A =	**90 – 100%**
B =	**80 – 89%**
C =	**70 – 79%**
D =	**60 – 69%**
F =	**0 – 59%**

Course Policies

ATTEND ZOOM CLASS
Participate WHEN IN CLASS
Keep up with the assignments, keep up with assignments, keep up with assignments
Self-motivation is a must!
Do your reading before the next class session. Attend office hours and/or text when you can make it

Academic Integrity

There are many different forms of academic dishonesty. The following kinds of honesty violations and their definitions are not meant to be exhaustive. Rather, they are intended to serve as examples of unacceptable academic conduct.

- Plagiarism is taking and presenting as one's own the writings or ideas of others, without citing the source. You should understand the concept of plagiarism and keep it in mind when taking exams and preparing written materials. If you do not understand how to "cite a source" correctly, you must ask for help.

- Cheating is defined as fraud, deceit, or dishonesty in an academic assignment, or using or attempting to use materials, or assisting others in using materials that are prohibited or inappropriate in the context of the academic assignment in question.

Anyone caught cheating or plagiarizing will receive a zero (0) on the exam or assignment, and the instructor may report the incident to the Campus Disciplinary Officer, who may place related documentation in a file. Repeated acts of cheating may result in an F in the course and/or disciplinary action. Please refer to the General Catalog for more information on academic dishonesty or other misconduct. Acts of cheating include, but are not limited to, the following: (a) plagiarism; (b) copying or attempting to copy from others during an examination or on an assignment; (c) communicating test information with another person during an examination; (d) allowing others to do an assignment or portion of an assignment; (e) using a commercial term paper service.

How do I show academic honesty and integrity in an online "classroom"?

- KEEP YOUR PASSWORDS CONFIDENTIAL. ○ You have a unique password to access online software like Canvas. Never allow someone else to log-in to your account.

- COMPLETE YOUR OWN COURSEWORK. ○ When you register for an online class and log-in to Canvas, you do so with the understanding that you will produce your own work, take your own exams, and will do so without the assistance of others (unless directed by the instructor).

Examples of Academic Dishonesty that can occur in an online environment:

- Copying from others on a quiz, test, examination, or assignment;
- Allowing someone else to copy your answers on a quiz, test, exam, or assignment;
- Having someone else take an exam or quiz for you;
- Conferring with others during a test or quiz (if the instructor didn't explicitly say it was a group project, then he/she expects you to do the work without conferring with others);
- Buying or using a term paper or research paper from an internet source or other company or taking any work of another, even with permission, and presenting the work as your own;
- Excessive revising or editing by others that substantially alters your final work;
- Sharing information that allows other students an advantage on an exam (such as telling a peer what to expect on a make-up exam or prepping a student for a test in another section of the same class);
- Taking and using the words, work, or ideas of others and presenting any of these as your own work is plagiarism. This applies to all work generated by another, whether it be oral, written, or artistic work. Plagiarism may either be deliberate or unintentional.

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

Date or Week	Activity, Assignment, and/or Topic	Pages/ Due Dates/Tests
Week 1 Feb 16 – 19	**Fundamentals** • Printing • Data Types • User Input • Arithmetic • Modulus	**Chapter 1 & 2 (pg. 1-65)**
Week 2 Feb 22 – 26	**Fundamentals & Selection** • `Math.random()` • `if`-statements • `AND`/`OR` operators • `NOT` operator • `MOD` & `if`-statements • `if-else` statements	**Chapter 3 (pg. 75-106)**
Week 3 Mar 1 – 5	**Selection & Repetition** • `if-else if` statements • `while` loops • `do-while` loops • `for` loops	**Chapter 4 & 5 (pg. 119-191)**
Week 4 Mar 8 – 12	**Programming Application (Project 1)**	
Week 5 Mar 15 – 19	**Repetition & Arrays** • 1-D arrays • `for` loops & arrays • `for-each` loops & arrays • 2-D arrays	**Chapter 6 & 7 (pg. 205-274)**
Week 6 Mar 22 – 26	**2-D Arrays & ArrayLists** • Nested `for` loops & 2-D arrays • ArrayList ○ **`add()`** ○ **`remove()`** ○ **`get()`** ○ **`isEmpty()`** ○ **`contains()`** ○ **`etc…`**	**Chapter 8 (pg. 289-303)**

Date or Week	Activity, Assignment, and/or Topic	Pages/ Due Dates/Tests
Week 7 Mar 29 – April 2	**Programming Application (Project 2)**	
Week 8 April 5-9	**Spring Break**	
Week 9 April 12 – 17	**Encapsulation** • Functions & Methods • Parameters • Return type • Private/Public	**Chapter 9-10 (pg. 323-394)**
Week 10 April 19 – 23	**Encapsulation & Class Design** • Fields • Constructors • The `this` reference • Accessor & Mutator Methods • `toString()` method	
Week 11 April 26 – 30	**Class Design** • Class Instantiation • Zero & multiple argument Constructor • Overloading methods • Static vs. non-static methods & fields	
Week 12 May 3 – 7	**Class Design** • Passing Object to Methods • Array of Objects • String class **Programming Application (Project 3)**	
Week 13 May 10 – 14	**Inheritance** • `extends` keyword • Superclasses & subclasses • `super` keyword • Overriding and overloading methods	**Chapter 11 (pg. 411-445)**
Week 14 May 17 – 21	**Interfaces** • Abstract classes • `implements` keyword • overriding abstract methods	**Chapter 13 (pg. 499-531)**
Week 15 May 24 – 28	**Programming Application (Project 4)**	
Week 16	**GUI**	**Chapter 14 (pg. 541-580)**

Date or Week	Activity, Assignment, and/or Topic	Pages/ Due Dates/Tests
June 1 –4	• JavaFX vs. Swing and AWT • JavaFX basic structure • Color class • Font class • Panes & Groups	
Week 17 June 7 – 11	**Final** **Programming Application & Exam (Final)**	

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