CIC Approval: 05/12/2022
BOT APPROVAL:
STATE APPROVAL:
EFFECTIVE TERM: Spring 2023

SAN DIEGO COMMUNITY COLLEGE DISTRICT
MESA, AND MIRAMAR COLLEGES
ASSOCIATE DEGREE COURSE OUTLINE

SECTION I

SUBJECT AREA AND COURSE NUMBER: Computer and Information Sciences 191

COURSE TITLE:
Intermediate Java Programming

Units:
4
Grade Only

CATALOG COURSE DESCRIPTION:

This course is an intermediate level study of the Java programming language. Topics include single and multidimensional arrays; objects and classes; object-oriented programming; inheritance and polymorphism; exception handling and text input/output (I/O); abstract classes and interfaces; graphical user interfaces (GUIs); recursion; concurrency; and generic collections and data structures, such as linked lists, queues, and stacks. This course is intended for students majoring in computer and information sciences or anyone interested in learning more about the Java programming language.

REQUISITES:

Prerequisite:
CISC 190 with a grade of "C" or better, or equivalent

FIELD TRIP REQUIREMENTS:
May be required

TRANSFER APPLICABILITY:
Associate Degree Credit & transfer to CSU UC Transfer Course List

CID:
COMP 132

TOTAL LECTURE HOURS:
48 - 54

TOTAL LAB HOURS:
48 - 54

TOTAL CONTACT HOURS:
96 - 108

OUTSIDE-OF-CLASS HOURS:
96 - 108

TOTAL STUDENT LEARNING HOURS:
192 - 216

STUDENT LEARNING OBJECTIVES:

Upon successful completion of the course the student will be able to:

1. Employ design principles of object-oriented programming.
2. Utilize the syntax and semantics of an object-oriented language in the development of software.
3. Effectively use software development tools including libraries, compilers, editors, debuggers, and version control systems.
4. Construct programs utilizing single and multidimensional arrays; objects and classes; object-oriented programming; inheritance and polymorphism; exception handling and text I/O; abstract classes and interfaces; GUI; recursion; concurrency, and generic collections and data structures, such as linked lists, queues, and stacks.
5. Systematically design and implement intermediate-level Java programs from given specifications, such as problem descriptions, user stories, design criteria, input/output scenarios, functional or unit tests, pseudo code, modeling language specifications or diagrams.
6. Articulate the purpose and working of a program, and its language constructs and data structures, as included in this course, such as through diagrams (for instance, flowcharts, Unified Modelling Language (UML) diagrams or entity-relationship (ER) diagrams), code comments, screencast videos, interactive computing notebooks (for instance, Jupyter notebooks), or written or oral communication.
7. Document external interfaces of intermediate-level Java methods (the application programming interface (API); for instance, Javadoc), and the inner working of methods in code comments.
8. Systematically test and debug intermediate-level Java programs by interactively entering arguments, utilizing functional or unit tests, running programs in a test environment, or using a debug environment (for instance, a debugger in an integrated development environment (IDE)).
9. Develop intermediate-level Java programs using industry recognized tools, such as code editors, command lines (Read, execute, print loop, REPL), emulators, or integrated development environments (IDE).
10. Create a fully functional, interactive program incorporating a substantial set of topics from the course.

SECTION II

1. COURSE OUTLINE AND SCOPE:

A. **Outline Of Topics:**
The following topics are included in the framework of the course but are not intended as limits on content. The order of presentation and relative emphasis will vary with each instructor.

I. Review of arrays
  A. Single-dimensional
    1. Copying arrays
    2. Passing arrays to methods
    3. Returning an array from a method
    4. Variable-length argument lists
    5. Search arrays
    6. Sorting arrays
    7. The array class
    8. Command-line arguments
    9. Ragged arrays
  B. Multidimensional
    1. Processing two-dimensional arrays
    2. Passing two-dimensional arrays to methods
    3. Other aspects of multidimensional arrays
II. Review of objects and classes
  A. Defining classes for objects
  B. Constructing objects using constructors
  C. Accessing objects via reference variables
  D. Using classes from the Java library
  E. Static variables, constants, and methods
  F. Visibility modifiers
  G. Data field encapsulation
  H. Passing objects to methods

I. Array of objects
J. Immutable objects and classes
K. The scope of variables
L. The "this" reference

III. Object-oriented programming
A. Class abstraction and encapsulation
B. Thinking in objects
C. Class relationships
D. Processing primitive data type values as objects
E. Automatic conversion between primitive types and wrapper class type
F. The BigInteger and BigDecimal classes
G. The String class
H. The StringBuilder and StringBuffer classes

IV. Inheritance and polymorphism
A. Superclasses and subclasses
B. Use of the "super" keyword
C. Overriding methods
D. Overriding vs. overloading
E. The object class and its toString() method
F. Polymorphism
G. Dynamic binding
H. Casting objects and the "instanceof" operator
I. The object's "equals" method
J. The ArrayList class
K. Useful methods for lists
L. The "protected" data and methods
M. Preventing extending and overriding

V. Exception handling and text I/O
A. Exception types
B. The "finally" clause
C. When to use exceptions
D. Rethrowing exceptions
E. Chained exceptions
F. Defining custom exception classes
G. The File class
H. File input and output
I. Reading data from the Web

VI. Abstract classes and interfaces
A. Abstract classes
B. Interfaces
C. The comparable interface
D. Interfaces vs. abstract classes
E. Class design guidelines

VII. GUI basics
A. JavaFX vs. Swing vs. Abstract Window Toolkit (AWT)
B. Basic structure of a GUI program: Model-View-Controller architecture
C. Panes, User Interface (UI) controls, and shapes
D. Property binding
E. Common properties and methods for notes
F. The Color class
G. The Font class
H. The Image and ImageView classes' layout panes
I. Shapes

VIII. Event-driven programming and animations
A. Events and event sources
B. Registering handlers and handling events
C. Inner classes
D. Anonymous inner class handlers
E. Simplifying event handling using lambda expressions
F. Mounts events

- G. Key events
- H. Listeners for observable objects
- I. Animation

IX. UI controls and multimedia
- A. Labeled and Label
- B. Button
- C. Checkbox
- D. RadioButton
- E. TextField
- F. TextArea
- G. ComboBox
- H. ListView
- I. ScrollBar
- J. Slider
- K. Video and Audio

X. Recursion
- A. Problem solving using recursion
- B. Recursive helper methods
- C. Recursion vs. iteration
- D. Tail recursion

XI. Sorting
- A. Insertion sort
- B. Bubble sort
- C. Merge sort
- D. Quick sort
- E. Heap sort

XII. Multithreading and parallel programming
- A. Thread concepts
- B. Creating tasks and threads
- C. The thread class
- D. Thread pools
- E. Thread synchronization
- F. Synchronization using locks
- G. Cooperation among threads
- H. Blocking queues
- I. Semaphores
- J. Avoiding deadlocks
- K. Thread states
- L. Synchronized collections
- M. Parallel programming

XIII. Memory management
- A. The stack
- B. The heap
- C. Garbage collection

XIV. Generic collections and data structures
- A. Iterators
- B. Array List
- C. Queue
- D. Linked List
- E. Stack
- F. Hash Table
- G. Trees

XV. Version control systems
- A. Benefits
  1. Collaboration
  2. Backups
  3. Documentation of changes
- B. Key operations
  1. Checkout
  2. Edit

3. Commit
4. Update

B. **Reading Assignments:**
Reading assignments are required and may include, but are not limited to, the following:

I. Course textbook(s).
II. Websites about the use of the Java language in technology, such as pages and articles on the Institute of Electrical and Electronics Engineers (IEEE) website.
III. Articles in professional journals or magazines, such as the Association of Computing Machinery (ACM) Journal.

C. **Writing Assignments:**
Writing assignments are required and may include, but are not limited to, the following:

I. Written explanations of Java programming tasks.
II. Written descriptions of concepts in the Java programming language, such as recursion vs. iteration.
III. Flowcharts detailing the steps and logical flow of a Java program.

D. **Appropriate Outside Assignments:**
Outside assignments may include, but are not limited to, the following:

I. Completing reading and writing assignments.
II. Evaluating and analyzing Java programming requirements.
III. Creating Java programs or program components.

E. **Appropriate Assignments that Demonstrate Critical Thinking:**
Critical thinking assignments are required and may include, but are not limited to, the following:

I. Constructing programs using the tools and features of Java.
II. Evaluating Java programs and logically debugging and correcting errors.
III. Utilizing the syntax and semantics of an object-oriented language in the development of software.

2. METHODS OF EVALUATION:

A student's grade will be based on multiple measures of performance unless the course requires no grade. Multiple measures may include, but are not limited to, the following:

I. Performance on hands-on programming assignments.
II. Written responses to in-class assignments.
III. Responses to in-class objective and/or essay question quizzes and/or examinations.
IV. Development of programs in Java.
V. Interactive one-on-one demonstration of program testing and operations in Java.
VI. Participation in classroom discussion.
VII. Development of Java program documentation.

3. METHODS OF INSTRUCTION:

Methods of instruction may include, but are not limited to, the following:

* Audio-Visual
* Collaborative Learning
* Computer Assisted Instruction
* Distance Education (Fully online)
* Lecture-Lab Combination
* Other (Specify)
* A. In-class computer hands-on practice of concepts and techniques included in course objectives.
* B. Interactive group activities including analysis, evaluation, and modification of Java programs.

4. REQUIRED TEXTS AND SUPPLIES:
Textbooks may include, but are not limited to:

TEXTBOOKS:
1. Farrell, Joyce. Java Programming, 9th ed. Cengage Learning, 2019, ISBN: 9781337397070
2. Gaddis, Tony. Starting Out with Java: From Control Structures through Objects, 7th ed. Pearson, 2019, ISBN: 9780134802213
3. Liang, Y. Daniel. Intro to Java Programming, Comprehensive Version, 12th ed. Pearson, 2019, ISBN: 9780136529153
4. Savitch, Walter. Absolute Java, 6th ed. Pearson, 2016, ISBN: 9780134041674

MANUALS:

PERIODICALS:

SOFTWARE:

SUPPLIES:

ORIGINATOR: Allan Schougaard

CO-CONTRIBUTOR(S)
DATE: 11/09/2021