Western Technical College

10152155 Java 2

Course Outcome Summary

Course Information

Description	The goal as programmers, is to create reliable, efficient, and sustainable solutions to problems. Java 2 introduces powerful design concepts and algorithms that programmers utilize to solve challenging problems congruent with these goals. The course starts with a brisk review of select topics from Java I and proceeds to more advanced design centric concepts. Using the Java programming language, Java 2 focuses on the more complex aspects of programming, design, data structures, algorithms, I/O, performance, and etc.
Career Cluster	Information Technology
Instructional Level	Associate Degree Courses
Total Credits	4
Total Hours	90

Textbooks

Java: An Introduction to Problem Solving and Programming - with Access. 8th Edition. Copyright 2018. Savitch, Walter. Publisher: Pearson. **ISBN-13**: 978-0-13-446203-5. Required.

Course Competencies

1. Review data types, objects, classes, and control structures.

Assessment Strategies

1.1. Lab/Project
1.2. Written Product

Criteria

You will know you are successful when

1.1. you complete a lab which demonstrates working with primitive and object data types.
1.2. you complete a lab which demonstrates working with arrays.
1.3. you complete a lab which demonstrates working with instance variables, constructors, and methods.
1.4. you complete a lab which demonstrates utilizing loops and conditionals.
1.5. you complete a written assessment where you write short code snippets to demonstrate understanding of primitives, objects, arrays, classes, loops, and conditionals.

Learning Objectives

1.a. Declare primitives and instantiate object data types.
1.b. Work with 1D and 2D arrays.
1.c. Define and instantiate an encapsulated Java class which includes instance variables, constructors, and

methods.

1.d. Utilize loops and conditionals.

2. Examine object inheritance, polymorphism, and interfaces.

Assessment Strategies

2.1. Lab/Project
2.2. Written Product

Criteria

You will know you are successful when:

2.1. you complete a lab which demonstrates the ability to define and utilize inheritance and polymorphism.
2.2. you complete a lab which demonstrates the ability to define and utilize interfaces and derived classes.
2.3. you complete a written assessment where you write short code snippets to demonstrate understanding of inheritance and polymorphism.
2.4. you complete a written assessment where you write short code snippets to demonstrate understanding of interfaces and derived classes.

Learning Objectives

2.a. Utilize inheritance and polymorphism.
2.b. Define and implement interfaces.
2.c. Define and implement derived classes.

3. Utilize exception handling.

Assessment Strategies

3.1. Lab/Project
3.2. Written Product

Criteria

You will know you are successful when

3.1. you complete a lab which demonstrates working with delivered exception classes.
3.2. you complete a lab which demonstrates implementing custom exception classes.
3.3. you complete a lab which demonstrates an understanding of the mechanics of exception handling; specifically when, where, and how, to throw and catch exceptions.
3.4. you complete a written assessment where you define, throw, and catch exceptions.
3.5. you complete a written assessment where you rationalize exception handling.

Learning Objectives

3.a. Define the concept of and the need for exception handling.
3.b. Understand the mechanics of exception handling and react correctly when particular exceptions occur.
3.c. Implement custom exceptions.

4. Explore file input and output.

Assessment Strategies

4.1. Lab/Project
4.2. Written Product

Criteria

You will know you are successful when

4.1. you complete a lab which demonstrates reading from and writing to a text file.
4.2. you complete a lab which demonstrates reading from and writing to a binary file.
4.3. you complete a lab which demonstrates the ability to parse and process data from a file.
4.4. you complete a lab which demonstrates appropriate error/exception handling while working with I/O.
4.5. you complete a written assessment where you demonstrate reading, writing, and processing information via text and binary files.

Learning Objectives

4.a. Define the concept of an I/O stream.
4.b. Define and identify the differences between a text and binary file.
4.c. Write and save binary and text data to a file.
4.d. Read binary and text data from a file.

5. Apply recursive concepts and techniques.

Assessment Strategies

5.1. Lab/Project
5.2. Written Product

Criteria

You will know you are successful when

5.1. you complete a lab which includes creating a binary search algorithm.
5.2. you complete a lab which includes creating a merge sort algorithm.
5.3. you complete a written assessment which demonstrates an understanding of recursive concepts.
5.4. you complete a written assessment which demonstrates an understanding of a binary search recursive algorithm.
5.5. you complete a written assessment which demonstrates an understanding of a merge sort algorithm.
5.6. you complete a written assessment which demonstrates an understanding of a quick sort algorithm.

Learning Objectives

5.a. Utilize recursive techniques to solve problems.
5.b. Utilize recursion as a programming tool.
5.c. Explore common uses of recursion.
5.d. Examine and implement binary search and merge sort recursive algorithms.

6. Utilize abstract data types.

Assessment Strategies

6.1. Lab/Project
6.2. Written Product

Criteria

You will know you are successful when

6.1. you complete a lab which demonstrates the ability to utilize several of Java’s delivered data structures; ArrayLists, Vector, HashMap, and TreeMap.
6.2. you complete a lab which demonstrates the ability to implement and manipulate a custom linked list data structure.
6.3. you complete a lab which utilizes inner classes, generics, and iterators.
6.4. you complete a written assessment demonstrating the ability to utilize Java’s build in array-based and collection-based data structures.
6.5. you complete a written assessment demonstrating the ability to implement and manipulate a custom linked list data structure.

Learning Objectives

6.a. Define the concept of an abstract data type (ADT).
6.b. Explain the need for data structures in programming.
6.c. Explore and utilize some of Java’s delivered array-based data structures; ArrayList, Vector, HashMap, and TreeMap.
6.d. Explore and utilize some of Java’s delivered data structures from the Java Collections Framework; HashSet, HashMap, and TreeMap.
6.e. Describe, implement, and manipulate linked list data structures.
6.f. Utilize inner classes, generic types, and iterators.

7. Examine searching and sorting algorithms.

Assessment Strategies

7.1. Lab/Project
7.2. Written Product

Criteria

You will know you are successful when

7.1. you complete labs implementing the following search algorithms: linear search and binary search.
7.2. you complete a written assessment demonstrating an understanding of the following search algorithms: linear search, binary search.

7.3. you complete labs implementing the following sortation algorithms: bubble sort, insertion sort, selection sort, quick sort, merge sort.
7.4. you complete a written assessment demonstrating an understanding of the following sortation algorithms: bubble sort, insertion sort, selection sort, quick sort, merge sort.

Learning Objectives

7.a. Analyze and implement the following fundamental search algorithms: linear search, binary search.
7.b. Analyze and implement fundamental sorting algorithms: bubble sort, insertion sort, selection sort, quick sort, merge sort.

8. Explore Big O notation and performance concepts

Assessment Strategies

8.1. Written Product

Criteria

You will know you are successful when:

8.1. you complete a written assessment demonstrating the ability to derive the growth function of an algorithm.
8.2. you complete a written assessment demonstrating the ability to simplify the growth function of an algorithm in terms of Big O notation.
8.3. you complete a written assessment demonstrating the ability to differentiate between growth functions.
8.4. you complete a written assessment demonstrating the ability to identify the common growth functions.
8.5. you complete a written assessment demonstrating the ability to describe several search and sort algorithms in Big O notation.

Learning Objectives

8.a. Derive and identify typical, worst, and best case performance of several algorithms using Big O notation.
8.b. Analyze performance to aid in choosing the best fit algorithm.

9. Explore building graphical user interfaces.

Assessment Strategies

9.1. Lab/Project

Criteria

You will know you are successful when:

9.1. you will complete several labs which require you to design a GUI.
9.2. you will complete several labs which require you to build a GUI.
9.3. you will complete several labs which accept and respond to events.
9.4. you will complete at least one lab which demonstrates creating a thread safe GUI.
9.5. you will complete several labs which explore different Java based GUI technologies.

Learning Objectives

9.a. Design and construct simple user interfaces utilizing Java.
9.b. Construct programs which respond to events from the GUI.
9.c. Utilize threading as it relates to user interface design.