Western Technical College

10152153 Introduction to Java

Course Outcome Summary

Course Information

Description	This course covers theoretical concepts and basic features of the Java programming language. Topics covered include data types, operators, control structures, arrays, Java functions, user-defined functions and object-oriented principles. Students will learn to develop the skill of formulating problem solution steps and translate that solution into Java code. Creating programs with graphical user interfaces and writing applets is also covered.
Career Cluster	Information Technology
Instructional Level	Associate Degree Courses
Total Credits	3
Total Hours	72

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.

Success Abilities

1. Cultivate Passion: Enhance Personal Connections
2. Cultivate Passion: Expand a Growth-Mindset
3. Cultivate Passion: Increase Self-Awareness
4. Live Responsibly: Develop Resilience
5. Live Responsibly: Foster Accountability
6. Refine Professionalism: Act Ethically
7. Refine Professionalism: Improve Critical Thinking

8. Refine Professionalism: Participate Collaboratively
9. Refine Professionalism: Practice Effective Communication

Course Competencies

1. Identify the key features of the Java Programming language.

Assessment Strategies

1.1. Written Product

Criteria

You will know you are successful when

1.1. you write short Java programs (approx. 500 lines or less).
1.2. you compile and run Java programs using the command-line and an IDE.
1.3. you explain Java's JVM, garbage collection, and syntax.
1.4. you produce code snippets to solve problems.
1.5. you use the Java Application Programming Interface (API) and other online resources.

Learning Objectives

1.a. Write short programs.
1.b. Compile and run Java programs from the command-line and an IDE.
1.c. Describe Java's JVM, portability, garbage collection, syntax, and modern usage.

2. Explore computing architecture and programming fundamentals.

Assessment Strategies

2.1. Written Product

Criteria

You will know you are successful when

2.1. you describe core computer components and their purpose/function.
2.2. you define the terms bit and byte.
2.3. you convert between decimal and binary number systems.
2.4. you describe the differences between an interpreted and compiled language.
2.5. you produce Java code snippets and Java projects that solve problems.
2.6. you use the Scanner and JOptionPane classes to interface with users.

Learning Objectives

2.a. Describe the purpose of the CPU, primary memory, and secondary memory.
2.b. Identify the processes involved when running a program.
2.c. Convert binary numbers to decimal and decimal numbers to binary.
2.d. Contrast interpreted and compiled languages.
2.e. Formulate basic algorithms.
2.f. Obtain user input from the keyboard.

3. Perform basic computations using variables and data types.

Assessment Strategies

3.1. Written Product

Criteria

You will know you are successful when

3.1. you declare and write assignment statements using all Java primitive types.
3.2. you apply the order of execution.
3.3. you use parenthesis in assignment statements.
3.4. you explain Java operators.
3.5. you declare named constants.
3.6. you use the String class to manipulate text.

Learning Objectives

3.a. Declare and write assignment statements using number and character data types.
3.b. Define named constants.
3.c. Manipulate strings of characters.

4. Apply industry standard best coding practices.

Assessment Strategies

4.1. Written Product

Criteria

You will know you are successful when

4.1. you use comments, correct case, indentation, and other formatting rules in their Java programs.
4.2. you describe the importance of writing quality code.
4.3. you demonstrate basic usage of a version control system.
4.4. you use classes, methods, and other code structures that facilitate code reuse.

Learning Objectives

4.a. Write Java programs using comments, correct casing, indentation, and other formatting rules.
4.b. Demonstrate coding for software reuse and sustainability.
4.c. Commit and checkout a program to a version control system.

5. Use code control structures.

Assessment Strategies

5.1. Written Product

Criteria

You will know you are successful when

5.1. you control program flow using if-else statements.
5.2. you control program flow using the switch statement.
5.3. you use nested and multibranch statements.
5.4. you evaluate Boolean operator precedence.
5.5. you iterate using for, while, and do-while loops.
5.6. you use the break and continue statements in loops.
5.7. you debug code control structures.

Learning Objectives

5.a. Use branching statements.
5.b. Compare primitive and object values.
5.c. Iterate using for, while, and do loops.

6. Apply fundamental object-oriented principles.

Assessment Strategies

6.1. Written Product

Criteria

You will know you are successful when

6.1. you describe the terms class, object, attribute, method, and constructor.
6.2. you write code to define a method.
6.3. you use the static keyword.
6.4. you use dot notation.
6.5. you instantiate objects.
6.6. you create custom Java classes.
6.7. you use the this keyword.
6.8. you encapsulate using the public and private keywords.
6.9. you write class constructors.
6.10. you use packages and the import statement.
6.11. you describe encapsulation, polymorphism, and inheritance.
6.12. you demonstrate subclassing an object.
6.13. you demonstrate knowledge of variable scope.
6.14. you demonstrate object comparison.

6.15. you create and use static variables and methods.
6.16. you explain how Java is a pass-by-value language.

Learning Objectives

6.a. Understand the concept of a class, instantiation of an object, and draw basic UML class diagrams.
6.b. Understand class, method, and block scope.
6.c. Construct methods and understand their mechanics --components of a method signature, parameter passing, preconditions and postconditions, overloading, and etc.
6.d. Encapsulate appropriately using public and private modifiers and create accessor and mutator methods.
6.e. Utilize object references, dot notation, the this keyword.
6.f. Perform object comparison.
6.g. Define static methods and variables.
6.h. Utilize packages and the import statement.
6.i. Apply the principles of encapsulation, polymorphism, and inheritance to design object-oriented programs.
6.j. Describe the basic components of a class – constructor, method, and field.

7. Explore debugging and testing a program.

Assessment Strategies

7.1. Written Product

Criteria

You will know you are successful when

7.1. you differentiate and resolve syntax errors, run-time errors, and logic errors.
7.2. you perform basic testing on code.
7.3. you use tools to assist in debugging a program.
7.4. you write a basic testing document.

Learning Objectives

7.a. Differentiate and resolve syntax errors, run-time errors, and logic errors.
7.b. Perform basic testing.
7.c. Utilize an IDE for debugging.

8. Examine fundamental data structures.

Assessment Strategies

8.1. Written Product

Criteria

You will know you are successful when

8.1. you construct single and multidimensional arrays.
8.2. you manipulate array data.
8.3. you implement basic sorting and searching algorithms using arrays.
8.4. you use arrays in classes and methods,

Learning Objectives

8.a. Construct single and multidimensional arrays.
8.b. Manipulate array data.
8.c. Implement basic sorting and searching algorithms using arrays.

9. Examine exception handling.

Assessment Strategies

9.1. Written Product

Criteria

You will know you are successful when

9.1. you explain the importance of exception handling.
9.2. you demonstrate usage of several kinds of exceptions and the exception hierarchy.
9.3. you demonstrate catching and throwing exceptions.
9.4. you explain the difference between data validation and exception handling.

Learning Objectives

9.a. Understand the importance of exception handling.
9.b. Understand the exception hierarchy in Java.
9.c. Differentiate between data validation and exception handling.
9.d. Catch and throw exceptions.

10. Explore working with various Java APIs.

Assessment Strategies

10.1. Written Product

Criteria

You will know you are successful when

10.1. you use the Math class.
10.2. you perform basic I/O.
10.3. you use the Wrapper classes.
10.4. you use the Java Application Programming Interface (API).

Learning Objectives

10.a. Explore the Math class.
10.b. Perform basic I/O.
10.c. Utilize the Wrapper classes.