Miami Dade College

Course Description

COP2800 | Java Programming | 4.00 credits

This is an intermediate level programming course using the Java computer language. Students will learn to code, compile, and execute programs while learning advanced programming concepts and object-oriented programming and design principles. Prerequisite: COP1047C, COP1334, or COP2270.

Course Competencies:

Competency 1: The student will demonstrate an understanding of the Java system architecture and its significant components by:

1. Distinguishing between the Java Runtime Environment (JRE) and the Java Development Kit (JDK)
2. Identifying the Java Virtual Machine (JVM) and the Java compiler
3. Describing the process of coding, compiling, and running from the command line
4. Differentiating between *.java and *.class files
5. Installing the JDK and compiling a program from the command line that uses at least one optional Java package
6. Using the JDK standard packages and API documentation in developing their programs

Competency 2: The student will demonstrate an understanding of the professional software development process by:

1. Designing and documenting solutions at the method level by writing pseudocode or developing flow charts for development before writing the code
2. Designing and documenting solutions at the project level using object-oriented design technology such as UML or CRC cards
3. Coding software solutions following professional coding style guidelines
4. Incorporating adequate and meaningful comments into the programming project source code using standard and JavaDoc style comments
5. Testing and designing tests of software solutions. Debugging program code

Competency 3: The student will demonstrate an understanding of fundamental programming constructs and concepts by:

1. Using appropriate data types for programming assignments. Using Boolean, comparison, arithmetic, and object (instance of) operators in their programs
2. Explaining the properties of a variable, such as its name, value, scope, persistence, and size
3. Distinguishing between expressions and statements
4. Identifying and using the three control structures (sequence, selection, and repetition)

Competency 4: The student will demonstrate an understanding of the following advanced programming techniques by:

1. Parsing a string and using other string manipulation techniques
2. Using both arrays and the Java collections to process aggregate data
3. Using object composition (object references) to build more complex objects
4. Developing an event-driven program
5. Writing a recursive algorithm for solving a problem and identifying its exit condition

Competency 5: The student will demonstrate an understanding of the object-oriented programming concepts of *Class* and *Object* by:

1. Identifying and using instance variables and instance methods
2. Using programming and identifying constructors
3. Explaining the process of object instantiation
4. Using programming and identifying accessor and mutator methods
5. Using programming and identifying class (static) variables and class (static) methods
6. Using programming and identifying overloaded methods and constructors
7. Creating programs using inner classes and describing their effects on generated class files

Updated: Fall 2025

Competency 6: The student will demonstrate an understanding of inheritance by:

1. Explaining the benefits of inheritance. Creating a class that extends a parent class
2. Explaining the restrictions imposed when using inheritance
3. Overriding and overloading parent class functions within a child class
4. Distinguishing between inheritance of implementation (extends) and inheritance of design (implements) Creating a class that implements an interface
5. Creating a class that extends an abstract class

Competency 7: The student will demonstrate an understanding of "Object Oriented Design." concepts by:

1. Using visibility modifiers (public, private, protected) to implement appropriate abstraction and encapsulation
2. Explaining coupling and how to achieve loose coupling. Explaining cohesion and how to achieve high cohesion
3. Writing a program that demonstrates polymorphism

Competency 8: The student will demonstrate an understanding of Java input and output by:

1. Describing I/O
2. Creating programs that use console I/O
3. Creating GUI (dialog box) I/O programs
4. Creating programs that use file I/O

Competency 9: The student will demonstrate an understanding of exception programming techniques by:

1. Describing exceptions. Encapsulating exceptions
2. Throwing and catching exceptions

<u>Learning Outcomes:</u>

- Use quantitative analytical skills to evaluate and process numerical data
- Solve problems using critical and creative thinking and scientific reasoning
- Use computer and emerging technologies effectively