

Module 2

JAVA Development - MADP 203

COURSE OUTLINE

COURSE DESCRIPTION:

This course for using Java is designed to be easy to use and is therefore easy to write, compile, debug, and learn than other programming languages. Java has ability to move easily from one computer system to another. The ability to run the same program on many different systems is crucial to World Wide Web software, and Java succeeds at this by being platform-independent at both the source and binary levels.

After this course candidates will be able to:

-
- Demonstrate the fundamentals of Java programming and how to use Java to write applications
 - Understand core object-oriented concepts,
 - Understand classes, packages, objects, methods, properties, abstraction, polymorphism, inheritance, encapsulation, and more
-

TOPIC OUTLINE / COURSE SCHEDULE:

Details provided on first day of class

Module/ Week/ Hours	Topic Details
Module 2/ Week 19-26/ 160 hours	A First Look <ul style="list-style-type: none">• A Simple Java Class• Java's "Hello World" Program Java Basics <ul style="list-style-type: none">• Language and Platform Features• Program Life Cycle• The Java SE Development Kit (JDK) Class and Object Basics <ul style="list-style-type: none">• The Object Model and Object-Oriented Programming• Classes, References, and Instantiation• Adding Data to a Class Definition• Adding Methods (Behavior)



Module/ Week/ Hours	Topic Details
	<p>More on Classes and Objects</p> <ul style="list-style-type: none">• Accessing data, the "this" variable• Encapsulation and Access Control, public and private Access• Constructors and Initialization• static Members of a Class• Scopes, Blocks, References to Objects <p>Flow of Control</p> <ul style="list-style-type: none">• Branching: if, if-else, switch• Iteration: while, do-while, for, break, continue <p>Strings and Arrays</p> <ul style="list-style-type: none">• String, String Buffer, String Builder• Arrays, Primitive Arrays, Arrays of Reference Types <p>Packages</p> <ul style="list-style-type: none">• Package Overview - Using Packages to Organize Code• import statements• Creating Packages, package Statement, Required Directory Structure• Finding Classes, Packages and Class path <p>Composition and Inheritance</p> <ul style="list-style-type: none">• Using Composition to Deal With Complexity• Composition/HAS-A, Delegation• Using Inheritance and Polymorphism to share commonality• IS-A, extends, Inheriting Features, Overriding Methods, Using Polymorphism• Class Object• Abstract Classes <p>Interfaces</p> <ul style="list-style-type: none">• Using Interfaces to Define Types• Interfaces and Abstract Classes <p>Exceptions</p> <ul style="list-style-type: none">• Exceptions and the Exception Hierarchy• try and catch• Handling Exceptions• Program Flow with Exceptions <p>JDBC</p> <ul style="list-style-type: none">• JDBC basics• JDBC Architecture

Module/ Week/ Hours	Topic Details
	<ul style="list-style-type: none"> • Using JDBC drivers & Driver Manager • Class Connection and connecting to a database • Class Statement and executing SQL statements • Other statement types • Driver types Java Collections and Generics <ul style="list-style-type: none"> • The Collections Framework and its API • Collections and Java Generics • Collection, Set, List, Map, Iterator • Auto boxing • Collections of Object (non-generic) • Using Array List, Hash Set, and Hash Map • for-each Loop • Processing Items With an Iterator • More About Generics Additional Java Features <ul style="list-style-type: none"> • Assertions • Type-safe Enums • Annotations • Additional Features in Java 6 and Java 7 • Conclusion Final Exam

COURSE CHANGES:

The content of this course outline is subject to changes. Material may be added or deleted from courses. Course content is revised on an ongoing basis to ensure relevance and accuracy of information. The instructor will notify students of changes as soon as possible. The course schedule may also need to be adjusted from time to times. In all cases content will be equal to or greater than the original.