Java Programming

OVERVIEW

This course teaches students how to develop Java applications. Topics covered include the Java programming language syntax, OO programming using Java, exception handling, file input/output, threads, collection classes, and networking. Students will develop and test Java applications (typically) using Eclipse. This course is a pre-requisite to all Application Server courses, and speciality Java Technology courses such as Struts, Spring, and Hibernate.

Course Contents:

- INTRODUCTION
- LANGUAGE COMPONENTS
- OBJECT-ORIENTED PROGRAMMING
- METHODS
- ARRAYS
- ENCAPSULATION
- INHERITANCE & POLYMORPHISM
- ABSTRACT CLASSES AND INTERFACES
- EXCEPTIONS
- INPUT AND OUTPUT IN JAVA
- COLLECTIONS
- NETWORKING
- THREADS

Course Content Details:

- **INTRODUCTION**
  1. What is Java?
  2. Versioning
  3. The Java Virtual Machine
  4. Writing a Java Program
  5. Packages
  6. Simple Java Programs
- **LANGUAGE COMPONENTS**
  1. Primitive Data Types
  2. Comments
  3. Control Flow Statements
  4. The `if` Statement
  5. The `switch` Statement
  6. The `while` and `do while` Statements
  7. The `for` Statement
  8. The `break` Statement
  9. The `continue` Statement
  10. Operators
  11. Casts and Conversions
  12. Keywords

Course Contents

- **OBJECT-ORIENTED PROGRAMMING**
  1. Defining New Data Types
  2. Constructors
  3. The `String` Class
  4. `String` Literals
  5. Documentation
  6. Packages
  7. The `StringBuffer` Class
  8. Naming Conventions
  9. The `Date` Class
  10. The `import` Statement
  11. Deprecation
  12. The `StringTokenizer` Class
  13. The `DecimalFormat` Class
- **METHODS**
  1. Introduction
  2. Method Signatures
  3. Arguments and Parameters
  4. Passing Objects to Methods
  5. Method Overloading
  6. Static Methods
  7. The `Math` Class
  8. The `System` Class
  9. Wrapper Classes
- **ARRAYS**
  1. Introduction
  2. Processing Arrays
  3. Copying Arrays
  4. Passing Arrays to Methods
  5. Arrays of Objects
  6. The Arrays Class
  7. Command Line Arguments
  8. Multidimensional Arrays
- **ENCAPSULATION**
  1. Introduction
  2. Constructors
  3. The `this` Reference
  4. Data Hiding
  5. `public` and `private` Members
  6. Access Levels
  7. Composition
  8. Static Data Members
- **INHERITANCE & POLYMORPHISM**
  1. Introduction
  2. A Simple Example
  3. The `Object` Class
  4. Method Overriding
  5. Polymorphism
  6. Additional Inheritance Examples
  7. Other Inheritance Issues
- **ABSTRACT CLASSES AND INTERFACES**
  1. Introduction
  2. Abstract Classes
  3. Abstract Class Example
  4. Extending an Abstract Class
  5. Interfaces
- **EXCEPTIONS**
  1. Introduction
  2. Exception Handling
  3. The Exception Hierarchy
  4. Checked Exceptions
  5. Advertising Exceptions with `throws`
  6. Developing Your Own Exception Classes
  7. The `finally` Block
- **INPUT AND OUTPUT IN JAVA**
  1. Introduction
  2. The `File` Class
  3. Standard Streams
  4. Keyboard Input
  5. File I/O Using Byte Streams
  6. Character Streams
  7. File I/O Using Character Streams
  8. Buffered Streams
  9. File I/O Using a Buffered Stream

Course Contents

10. Keyboard Input Using a Buffered Stream
11. Writing Text Files

- **COLLECTIONS**
  1. Introduction
  2. Vectors
  3. Hashtables
  4. Enumerations
  5. Properties
  6. Collection Framework Hierarchy
  7. Lists
  8. Sets
  9. Maps
  10. The `Collections` Class
- **NETWORKING**
  1. Networking Fundamentals
  2. The Client/Server Model
  3. `InetAddress`
  4. URLs
  5. Sockets
  6. A Time-of-Day Client
  7. Writing Servers
  8. Client/Server Example
- **THREADS**
  1. Threads vs. Processes
  2. Creating Threads by Extending `Thread`
  3. Creating Threads by Implementing `Runnable`
  4. Advantages of Using Threads
  5. Daemon Threads
  6. Thread States
  7. Thread Problems
  8. Synchronization