Syllabus for
Core JAVA

Course Duration for Core JAVA

- **6 weeks**

Objective For Core JAVA Course

- **Getting the student to be well trained in JAVA Programming**

Eligibility for Core JAVA Course

- **Any Technical Graduates or Undergraduate**
  (BSc, BCS, BCA, BE, B Tech, MSc, MCS, MCA, M Tech)

Course overview For Core JAVA Course

Core JAVA

Object Oriented Programming

- OOPS concepts and terminology
- Advantage of OOPS
- Fundamentals of OOPS

Introduction of JAVA

- What is JAVA?
- How to get JAVA
- A first JAVA program
- Compiling and interpreting applications
- The JDK Directory Structure
- Using Eclipse

Data types and Variables

- Variable names
- Primitive datatypes, declarations
- Numeric Literals, Character Literals
- String formatting and Parsing
- String Literals
- Arrays, non-primitive data types
- The Dot operator

Operators and Expressions

- Expressions
- Assignment Operator
- Arithmetic Operators
- Relational Operators
- Logical Operators
- Increment and Decrement Operators
- Operate-Assign Operators
- The Conditional Operator
- Operator Precedence
- Implicit Type Conversion
- The Cast Operator
- Strict Typing
- Type conversion

Control Flow Statements

- Statements
- Conditional (if) Statements
- Datatypes and Variables
- Adding an else if
- Conditional (switch) Statements
- While, do while loops, for loops
- A loop diagram
- Enhanced for Loop
- Continue statement and break statement

Syllabus for Core JAVA

GRADIENT INFOTECH

Object Oriented Programming

- Introduction to Object Oriented Programming
- Programming
- Classes and Objects
- Fields and methods
- Encapsulation
- Access Control
- Inheritance and Polymorphism

Methods

- Methods
- Calling methods
- Defining methods
- Method Parameters Scope
- So, why all the static?

Inheritance in JAVA

- Inheritance in JAVA
- Casting
- Methods Overriding
- Polymorphism
- Super keyword
- The Object Class

Packages

- The import statement
- Static imports
- Casting
- CLASSPATH and import
- Defining Packages
- Package Scope

Interface and Abstract Classes

- Separating interface and implementation
- UML interfaces and realizations
- Defining interfaces
- Implementing and Extending interfaces
- Runnable Threads
- Abstract Classes

Lambda Built-in Functional Interfaces

- Java.util function package
- Use primitive versions of functional interface
- Use binary versions of functional interface
- Use the Unary Operator Interface

Exception Handling

- Exceptions overview and catching exceptions
- Exception Methods
- Declaring,defining and throwing exceptions
- Errors and runtime Exceptions
- Assertions

Object and Class

- Defining a Class
- Creating an Object
- Instance Data and Class Data
- Methods
- Constructors
- Access Modifiers
- Encapsulation

Syllabus for
Core JAVA

GRADIENT INFOTECH

Collection Framework

- The Collection Framework
- The Set Interface
- Set Implementation Classes
- The List Interface
- List Implementation Classes
- The Map Interface
- Map Implementation Classes
- Utility Classes
- Generics
- Primitive wrapper Classes

Inner Class

- Inner Classes
- Member Classes
- Local Classes
- Anonymous Classes
- Instance Initializers
- Static Nested Classes

Introducing to Threads

- Non- Threaded Applications
- Threaded Applications
- Creating Threads
- Thread States
- Runnable Threads
- Co-ordinating Threads
- Interrupting Threads
- Runnable Interface, threads group

Project on Diploma in JAVA Course

Student will be assigned a project which they will have to execute under the guidance of the faculty.