

Programming With Java

UNIT 01: Introduction to Java
<ul style="list-style-type: none">○ Advantages of Java○ Compiling and running Java Program○ Basics of Object Oriented Programming○ Java Primitive Data Types○ Java Operators Set○ Primitive type Casting
UNIT 02: Control Statements
<ul style="list-style-type: none">○ if -else statement○ switch-case statement○ for Loop○ while loop○ do-while loop
UNIT 03: Declaring Class
<ul style="list-style-type: none">○ Basics of abstraction○ Class and Object Fundamentals○ Declaring class○ Instantiating class○ Java Memory Management○ Object and Reference○ Declaring Class Members (Methods and variables)○ Concept of Method overloading (Static polymorphism)
UNIT 04: Encapsulation and Java Modifiers
<ul style="list-style-type: none">○ Basics of encapsulation○ public, protected, private, default modifier○ Example for encapsulation○ Use of static keyword○ static methods and variables○ Declaring constant (final)○ Other Modifiers (strictfp, volatile, transient, abstract, final, extern)
UNIT 05: Constructors
<ul style="list-style-type: none">○ Declaring constructor○ Default constructor○ The this keyword○ Parameterized constructors○ Constructor chaining
UNIT 06: Packages
<ul style="list-style-type: none">○ Concept of package○ Use of API packages○ Creating own packages○ Using of user defined packages
UNIT 07: Declaring Arrays
<ul style="list-style-type: none">○ Declaring one dimensional array○ Instantiating and initializing array○ Array of arrays (Multi dimensional array)○ Passing arrays to methods○ Var-args

UNIT 08:Inheritance
<ul style="list-style-type: none"> ○ Basic concept of inheritance ○ Example of inheritance ○ Member access and Inheritance ○ Constructor in inheritance ○ Use of <code>super</code> keyword ○ Use of <code>super ()</code> constructor ○ Method overriding (Dynamic Polymorphism) ○ Dynamic method dispatch ○ Reference variable casting ○ <code>final</code> keyword ○ Abstract class ○ Example using abstract class ○ Interface ○ Difference between abstract class and interface ○ The <code>Object</code> class
UNIT 09:Garbage Collection
<ul style="list-style-type: none"> ○ Overview of java's garbage collection ○ Garbage collector overview ○ Writing code that explicitly makes object garbage collected
UNIT 10: Declaring enum constant
<ul style="list-style-type: none"> ○ Declaring enum ○ Example of enum
UNIT 11: Exception Handling
<ul style="list-style-type: none"> ○ What is an Exception in Java? ○ Exception class hierarchy ○ Handling Exception using <code>try, catch, finally</code> ○ Propagating exceptions ○ Use of <code>throw</code> and <code>throws</code> clause ○ Creating user's own exception class
UNIT 12: Handling String in Java
<ul style="list-style-type: none"> ○ The <code>String</code> class ○ Mutable and immutable objects ○ Important methods of the <code>String</code> class ○ The <code>StringBuffer</code> and <code>StringBuider</code> class ○ Important methods of <code>StringBuffer</code> and <code>StringBuider</code> class
UNIT 13: Wrapper Classes
<ul style="list-style-type: none"> ○ Overview of wrapper classes ○ Creating wrapper objects ○ Using wrapper conversion utilities ○ Concept auto-boxing and auto-unboxing ○ Method overloading revisited with wrapper objects and var-args
UNIT 14: I/O and File Handling
<ul style="list-style-type: none"> ○ Overview of file handling in java ○ Important classes of <code>java.io</code> package ○ Example of reading and writing into file ○ Object Serialization ○ The <code>java.io.Console</code> class

UNIT 15: Multithreaded Programming

- Overview of threading
- Defining, instantiating and running thread
- Thread state transition
- Preventing Thread execution
- Sleeping
- Thread priorities and `yield()`
- The `join()` method
- Synchronizing
- Inter Thread interaction

UNIT 16: Generics

- Overview of generics
- A simple example of generics
- How generics works?
- The general form of generic classes
- Generic classes with two and one type parameters
- Wild card parameters
- Creating generic methods
- Generic constructor
- Casting of generic references

UNIT 17: Collection Framework

- Overriding `toString()`, `hashCode()` and `equals()` methods
- Collection framework overview
- Importance classes and interfaces of collection framework
- Working with `List`
- Working with `Set`
- Working with `Map`
- Sorting and searching collections
- Generic collections
- Mixing generic and non-generic collections

UNIT 18: Inner classes

- Coding a regular inner class
- Accessing outer class members from inner class
- Method local inner class
- Anonymous inner class
- `static` nested classes