Object Oriented Programming Concepts

Sachchida Nand Chaurasia
Assistant Professor

Department of Computer Science
Banaras Hindu University
Varanasi
Email id: snchaurasia@bhu.ac.in, sachchidanand.mca07@gmail.com

August 12, 2025

CS 204	Object Oriented Programming	L	T	P
		3	0	2

Course Objectives

- Learn the Java programming language: its syntax, idioms, patterns, and styles
- Become comfortable with object oriented programming: Learn to think in objects
- Learn the essentials of the Java class library, and learn how to learn about other parts of the library when you need them
- Introduce event driven Graphical User Interface (GUI) programming

Detailed Syllabus I

- History of Java, Java Features, The three OOP principles: Encapsulation, Inheritance, and Polymorphism.
- **Data Types, Variables, and Arrays** : Data types, variables, constants, scope and life time of variables, operators, operator hierarchy, expressions, type conversion and casting, enumerated types, control flow-block scope, conditional statements, loops, break and continue statements, simple java program, arrays, input and output, formatting output

Detailed Syllabus II

- **Review of OOP concepts**: constructors, methods, static fields and methods, access control, this reference, overloading methods and constructors, recursion, garbage collection, building strings, exploring string class, Enumerations, Generics
- **Inheritance**: Inheritance concept, benefits of inheritance, Super classes and Sub classes, Member access rules, Inheritance hierarchies, super uses, preventing inheritance: final classes and methods, casting, polymorphism- dynamic binding, method

Detailed Syllabus III

overriding, abstract classes and methods, the Object class and its methods.

- **Interfaces**: Interfaces vs Abstract classes, defining an interface, implementing interfaces, accessing implementations through interface references, extending interface Inner classes –Uses of inner classes, local inner classes, anonymous inner classes, static inner classes, examples
- **Packages**: Defining, Creating and Accessing a Package, Understanding CLASSPATH, importing packages

Detailed Syllabus IV

- **Data structures creation and manipulation in java**: Introduction to Java Collections, Overview of Java Collection frame work, Commonly used Collection classes – ArrayList, LinkedList, HashSet, HashMap, TreeMap, Collection Interfaces – Collection, Set, List, Map, Legacy Collection classes – Vector, Hashtable, Stack, Dictionary(abstract), Enumeration interface, Iteration over Collections – Iterator interface, ListIterator interface Other Utility classes– StringTokenizer, Formatter, Random, Scanner, Observable, Using javautil

Detailed Syllabus V

- **Files streams**: byte streams, character streams, text Input/output, binary input/output, random access file operations, File management using File class, Using javaio
- **Exception handling**: Dealing with errors, benefits of exception handling, the classification of exceptions-exception hierarchy, checked exceptions and unchecked exceptions, usage of try, catch, throw, throws and finally, re-throwing exceptions, exception specification, built in exceptions, creating own exception sub classes, Guide lines for proper use of exceptions

Detailed Syllabus VI

- **Multithreading**: Differences between multiple processes and multiple threads, thread states, creating threads, interrupting threads, thread priorities, synchronizing threads, inter-thread communication, thread groups, daemon threads
- **GUI Programming with Java**: The AWT class hierarchy, Introduction to Swing, Swing vs AWT,MVC architecture, Hierarchy for Swing components, Containers – Top-level containers- Light weight containers – Overview of several swing components- Jbutton, JToggleButton, JCheckBox, JRadioButton,

Detailed Syllabus VII

JLabel, JTextField, JTextArea, JList, JComboBox, JMenu, Java's Graphics capabilities – Introduction, Graphics contexts and Graphics objects, color control, Font control, Drawing lines, rectangles and ovals, Drawing arcs, Layout management - Layout manager types – border, grid, flow, box

- **Event Handling** Events, Event sources, Event classes, Event Listeners, Relationship between Event sources and Listeners, Delegation event model, Semantic and Low-level events, Examples: handling a button click, handling mouse and keyboard events, Adapter classes

Detailed Syllabus VIII

- **Applets**: Inheritance hierarchy for applets, differences between applets and applications, life cycle of an applet - Four methods of an applet, Developing applets and testing, passing parameters to applets, applet security issues

Suggested Readings I

1.
2.
3.
4.
5.