



Object Oriented Programming

Chapter One: Introduction to Java Technology and Object Oriented Concepts

CHAPTER OUTLINE

- Java Overview and Brief History of java
- What is Java and Java Platform
- Java and Object Oriented Concepts Overview
- Simple Java class, Packages, Import Statements, Java Fields, Constructors and Methods
- A first Program in Java



Java Overview and Brief History of java

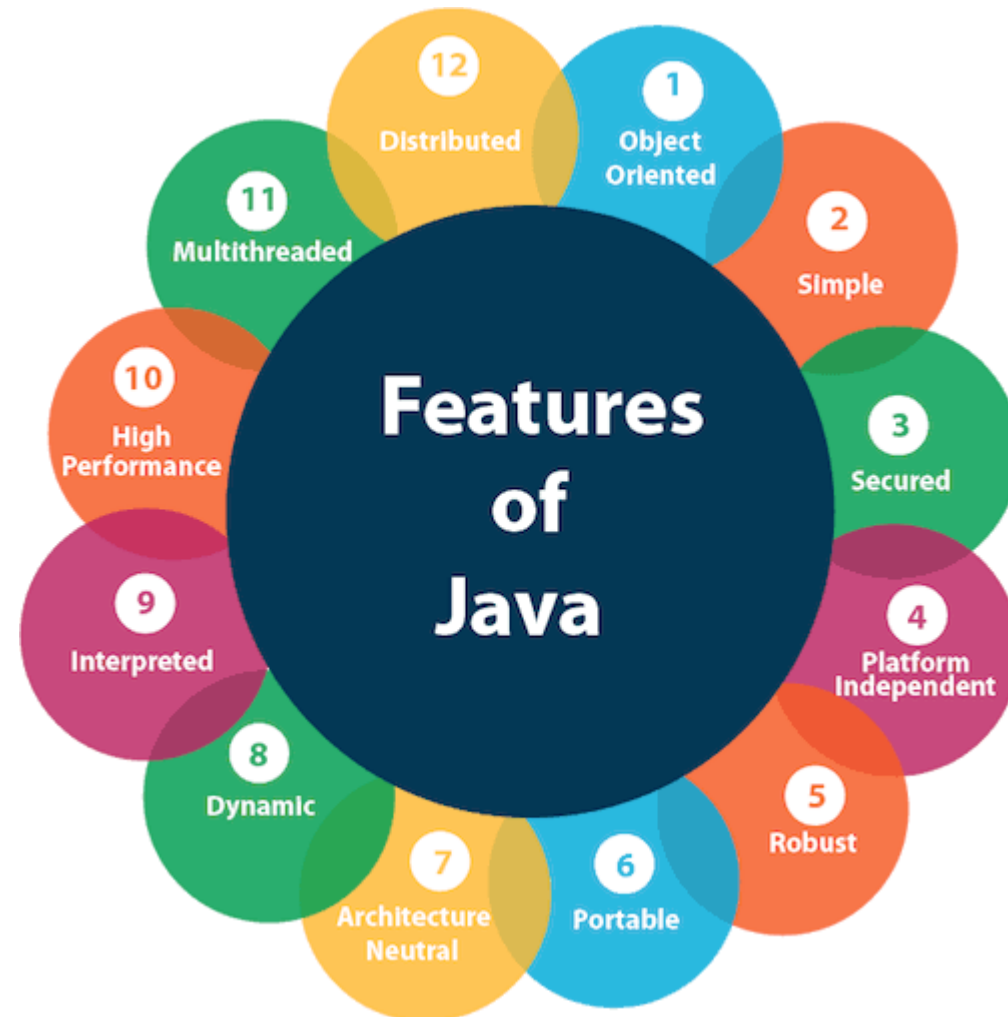
- The programming language you will be learning for this course is Java. Java is an object-oriented programming language developed by Sun Microsystems. Sun released the first version in May, 1995.
- Java was modelled after C++, the Java language was designed to be small, simple, and portable across platforms and operating systems.
- Java was written as a full-fledged programming language in which you can accomplish the same sorts of tasks and solve the same sorts of problems that you can in other programming languages, such as C or C++

Java Overview and Brief History of java

- There are **no pointers in Java**, nor is **there pointer arithmetic**. **Strings and arrays** are real objects in Java.
- Memory management is automatic.

What is Java and Java Platform

- Java technology is both a programming language and a platform.
- The Java programming language is a **high-level language** that can be characterized by all of the following buzzwords:
- **Portable, Secure, Simple, Object oriented, Distributed, Multithreaded, Dynamic.**



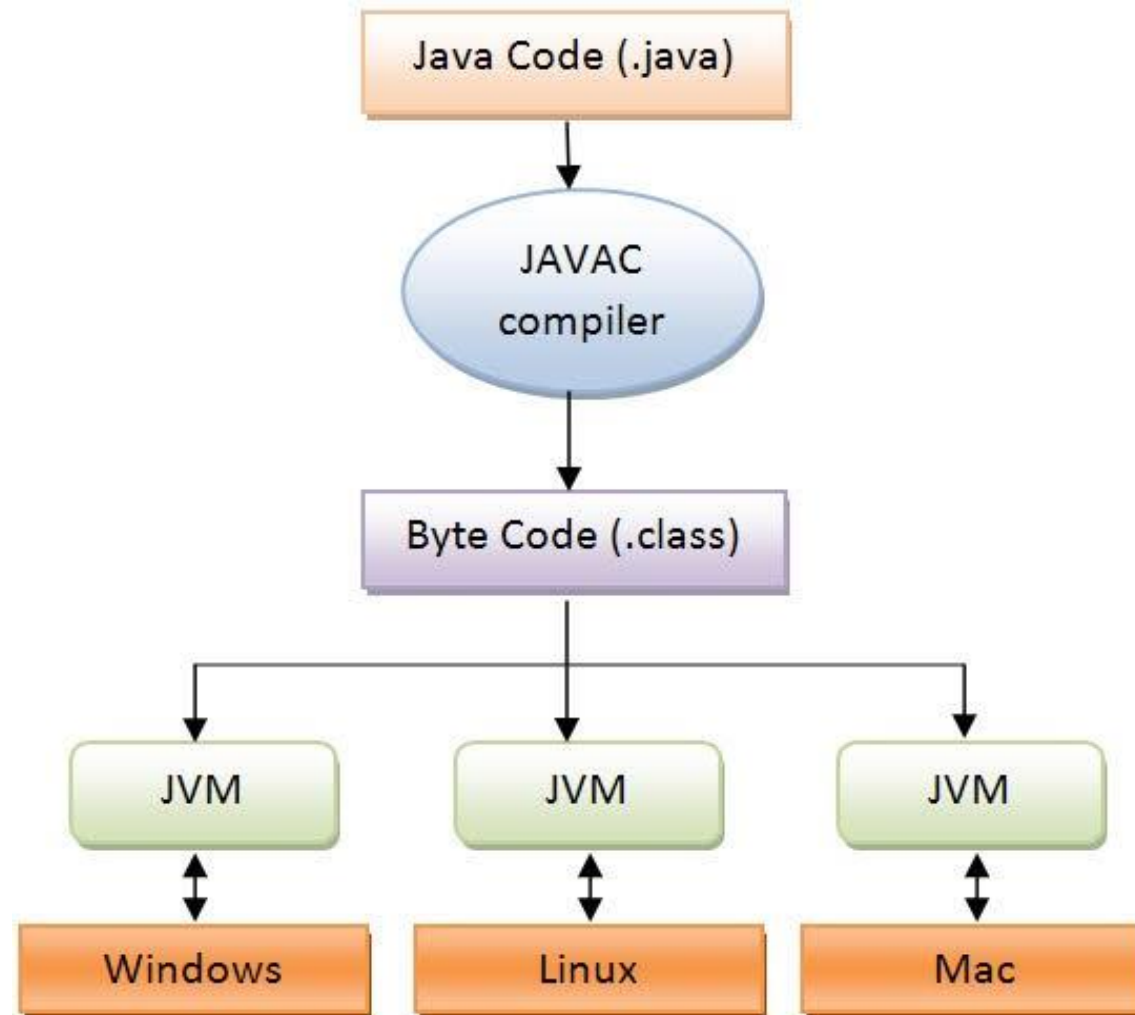
What is Java and Java Platform

- Java is different from other programming languages because a Java program is **compiled and interpreted**, in many other languages, the program is either compiled or interpreted.
- The Java compiler translates the program into an intermediate language called Java **byte codes**.
- Java byte codes are platform-independent. There are different **Java interpreters for different platforms**. The interpreter parses and runs each of the Java byte code instructions.
- Every Java interpreter is actually an implementation of something called the **Java VM (Virtual Machine)**.

What is Java and Java Platform

- In the Java programming language, all source code is first written in plain text files ending with the **.java extension**.
- Those source files are then compiled into **.class files** by the **javac** compiler.
- A **.class** file does not contain code that is native to your processor; it instead contains byte-codes - the machine language of the Java Virtual Machine (Java VM).
- The java launcher tool then runs your application with an instance of the Java Virtual Machine.

What is Java and Java Platform



What is Java and Java Platform

- A **platform** is the hardware or software environment in which a program runs.
- The most popular platforms are like Microsoft Windows, Linux, Solaris OS, and Mac OS. Most platforms can be described as a combination of the operating system and underlying hardware.
- The **Java platform** differs from most other **platforms** in that it's a software-only platform that runs on top of other hardware-based platforms. This is what makes it possible to **run compiled Java programs** on different platforms.
- The Java platform has two components: The **Java VM** and The Java **API (Application Programming Interface)**.

What is Java and Java Platform

- The API provides **libraries of pre-written**, ready-to-use components that provide standard functionality.
- A library groups a number of **related classes and interfaces** together into what is called a **Java package**.
- In order to write Java programs, you need to have the **Java SDK** (Software Development Kit) installed on your PC. This includes the **API and the VM**, as well as compilers and debuggers for Java.

Java and Object Oriented Concepts Overview

- Java is developed based on the **Object Oriented Paradigm or Concept**. The OO paradigm aims to eliminate some of the flaws of the procedural approach to programming.
- In **OO**, **data is critical**: data is not allowed to move freely around a system, but it is **tied closely to the functions** that operate on it. Data is also **protected from unintentional modification** by other functions.
- In OO programming, a system or problem is decomposed into entities called objects. An object has **data members and functions** : in Java, functions are known as **methods**. Data is protected because the data **belonging to an object** can only be accessed and modified by the **methods of that object**.

Java and Object Oriented Concepts Overview

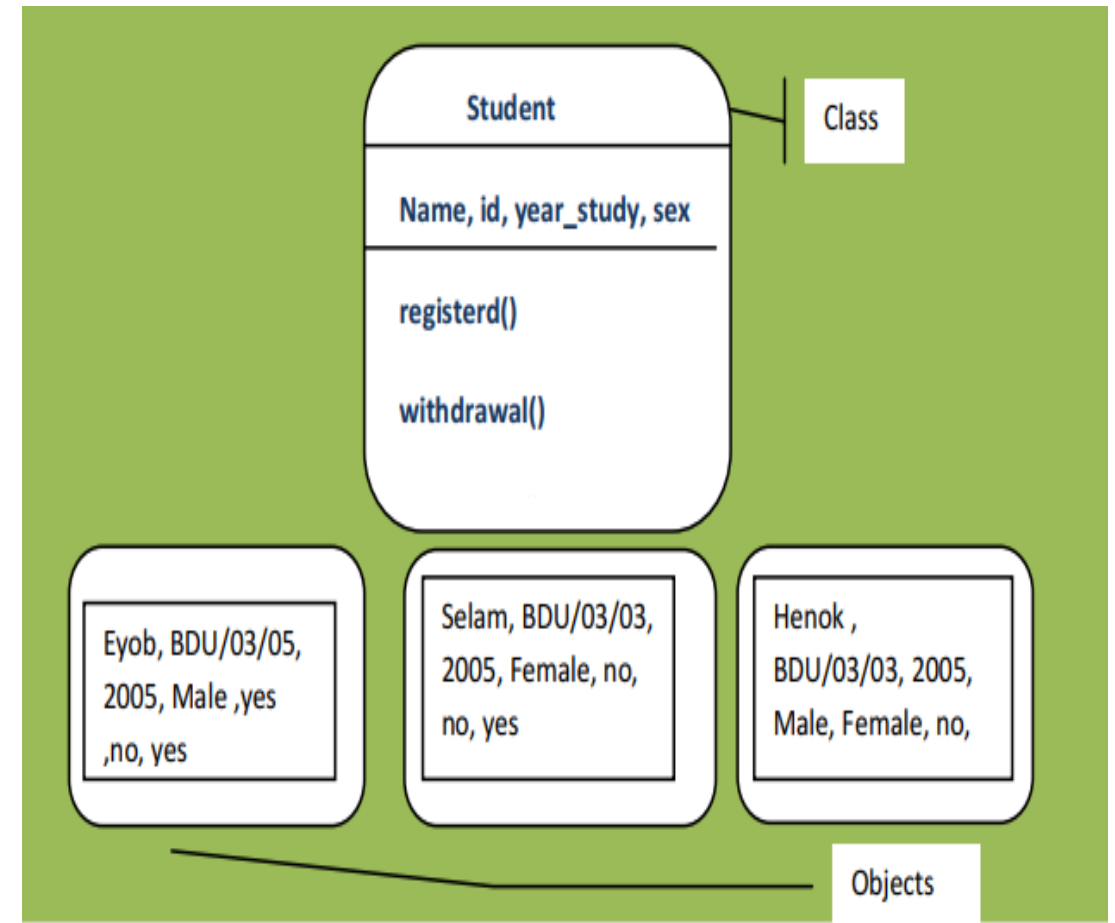
- **Object:** Objects are a key to understanding object-oriented technology.
- Look around right now and you'll find many examples of real-world objects: Class rooms, Students, Chalks, Black board and others.
- Real world objects share two characteristics: They all have **state and behaviour**. **Students** have state (name, age, sex, year of study) and behaviour (learning, eating, and studding).
- Identifying the state and behaviour for real-world objects is a great way to **begin thinking in terms of object oriented programming**.

Java and Object Oriented Concepts Overview

- Software objects are conceptually similar to real-world objects: they too consist of **state and related behaviours**.
- An object stores its **state in fields** (variables in some programming languages) and exposes its behaviours through **methods** (functions in some programming languages).
- Methods operate on an object's internal state and serve as the primary mechanism for object-to-object communication.

Java and Object Oriented Concepts Overview

- **Classes:** A class is the definition of a **set of objects** which are all similar in terms of their **attributes, associations and functionality**.
- **Example:** student class defines all student objects.
- A class is the blueprint from which individual objects are created.

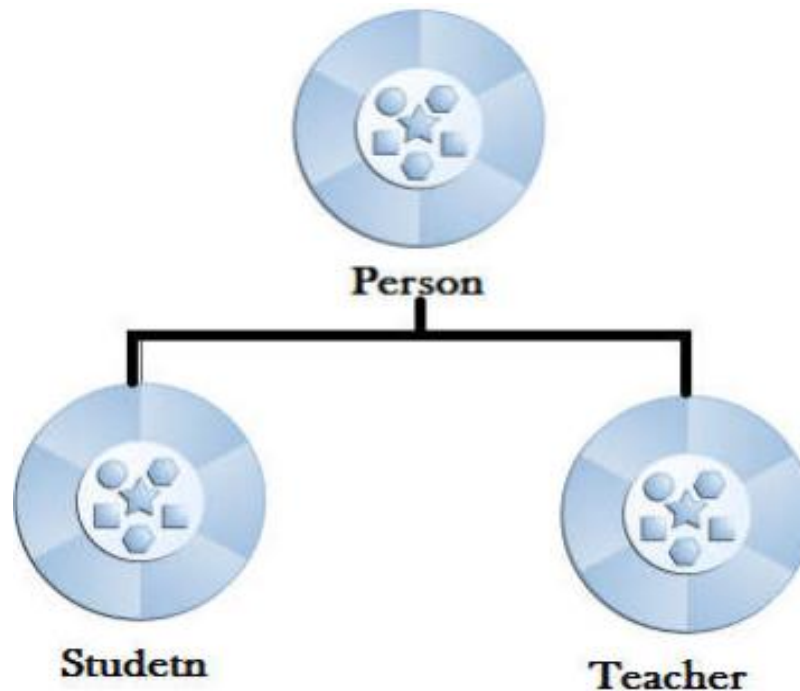


Java and Object Oriented Concepts Overview

- **Inheritance:** Inheritance is one of the fundamental mechanisms for code reuse in OOP. It allows new classes to be derived from an existing class.
- The new class (also called subclass, subtype, derived class, child class) can inherit members from the old class (also called super class, super type, base class, parent class).
- The subclass can add new behaviour and Properties and, under certain circumstances, modify its inherited behaviour.

Java and Object Oriented Concepts Overview

- In the Java programming language, each class is allowed to have **one direct superclass**, and each **superclass** has the potential for an **unlimited number of subclasses**.



Java and Object Oriented Concepts Overview

- **Package:** Packaging is a common approach used **to organize related classes and interfaces**.
- Packages are thought of as **containers for** classes, but actually they define where classes will be located in the hierarchical directory structure.
- Packaging is encouraged by Java coding standards to decrease the likelihood of classes colliding.

Java and Object Oriented Concepts Overview

- Packaging your classes also promotes **code reuse, maintainability**, and the **object oriented** principle of encapsulation and modularity.
- To place a source file into a package, use the **package** statement at the beginning of that file.
- The package statement includes the **package** keyword, followed by the package path delimited with periods.

Java and Object Oriented Concepts Overview

- **Import Statements:** An **import** statement allows you to include source code from other classes into a source file at **compile time**.
- To import classes from other packages into your source file, use the **import** statement.
- The **import** statement includes **import** keyword followed by the package path delimited with periods and ending with a **class name** or an **asterisk**.

Java and Object Oriented Concepts Overview

▪ **Class syntax:**

```
package <package_name>
import <other_package>
public class <class_name>{
    <fields or variables>;
    <constructors methods>{}
    <other methods>{}
}
```

Java and Object Oriented Concepts Overview

- If there is a **public class** in a file, the **name of the file** must match the name of the **public Class Name**.
- For example, a class declared as **public class Student { }** must be in a source code file named **Student.java**.
- There can be only **one public class** per source code file. A file can have more than **one non public class**.

Java and Object Oriented Concepts Overview

- **Variables:** or the data associated with the programmers (such as integers, strings, arrays, and references to other objects), are called instance fields or simply fields.
- **Constructors:** are functions called during the creation (instantiation) of object.
- **Methods:** are the functions that can be performed on object. They are also referred to as instance methods.

A first Program in Java

```
1
2  package lab1;
3
4  public class Welcome {
5
6      public Welcome(){}
7
8      public static void main(String[] args){
9          System.out.println("This is my first Object Oriented Java Program");
10     }
11 }
12
```