

Tutorial No. 1

Title: Basics of Java Programming.

Batch: SY-IT(B3)**Roll No. : 16010423076****Tutorial No.:1****Aim:** To study the basics of Java Programming

Resources needed: Java

Theory:

Object-Oriented Programming

Object-Oriented Programming (OOP) is a programming paradigm based on the concept of "objects", which can contain data and code to manipulate that data. The data is usually in the form of fields, often known as attributes or properties, and code, in the form of procedures, often known as methods.

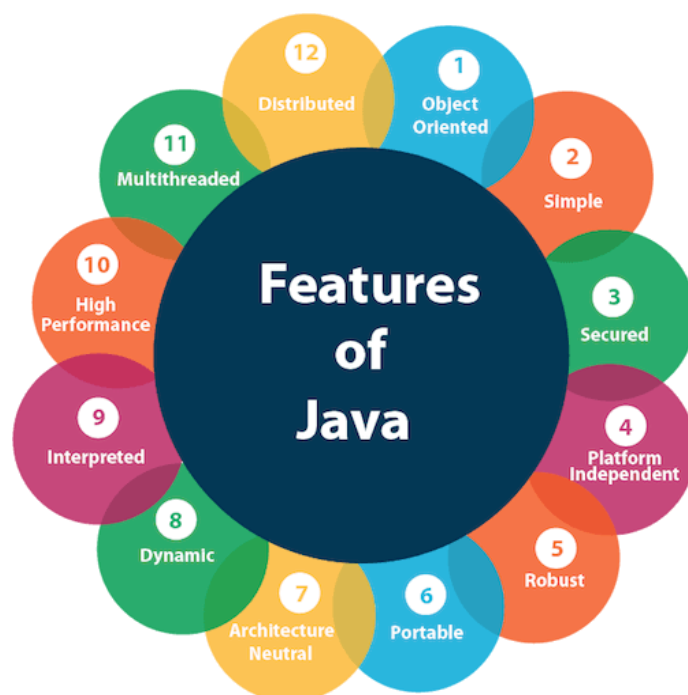
Features of Object-Oriented Programming:

1. **Encapsulation:**
 - **Definition:** Encapsulation is the mechanism of restricting direct access to some of an object's components and can prevent the accidental modification of data.
 - **Example:** Using private fields and providing public getter and setter methods to access and update the private fields.
2. **Abstraction:**
 - **Definition:** Abstraction is the concept of hiding the complex implementation details and showing only the necessary features of an object.
 - **Example:** An abstract class or an interface defines a blueprint without implementation details, which can be implemented by other classes.
3. **Inheritance:**
 - **Definition:** Inheritance is a mechanism where a new class is derived from an existing class. The new class inherits the properties and behavior of the existing class.
 - **Example:** A Bird class inherits from an Animal class, meaning Bird will have all the properties and methods of Animal.
4. **Polymorphism:**
 - **Definition:** Polymorphism allows methods to do different things based on the object it is acting upon. It can be achieved through method overloading and method overriding.
 - **Example:** A shape class may have a method draw(). Subclasses like Circle, Rectangle, and Triangle will implement the draw() method differently.
5. **Classes and Objects:**
 - **Class:** A blueprint for creating objects. It defines a datatype by bundling data and methods that work on the data.
 - **Object:** An instance of a class. It contains real values instead of variables.

What is Java?

Java is a programming language and a platform. Java is a high level, robust, object-oriented and secure programming language.

Java was developed by Sun Microsystems (which is now the subsidiary of Oracle) in the year 1995. James Gosling is known as the father of Java. Before Java, its name was Oak. Since Oak was already a registered company, so James Gosling and his team changed the name from Oak to Java.



Task: Prepare a document containing the answers of the following question

1. Explain the features of Java in detail.

Object-Oriented: It manages its code by means of objects. Therefore, organizing and reusing are easier. You can think about the objects as blocks, which model things in your program.

Platform-Independent: You can write a program in Java once and then run it on any device that has Java installed—on any type of computer or using any operating system.

Easy: Java is easy to learn and use. It eliminates the complex features found in other languages, hence being more straightforward for beginners.

Safe: Java has built-in features of safety that protect your computer from lousy code. It runs programs in an environment that is 'sandboxed' so that the programs cannot inflict damage.

Multithreaded: Java can process several tasks at one go in a program. This makes it great for developing fast and responsive applications. First and foremost, the language has been designed powerful and robust. It has automatic memory management and error handling, which helps in keeping the program run without disruption. Secondly, it is portable. Any Java program is easily portable across many types of computers and operating systems. This is possible because Java operates on a special kind of machine called JVM, which understands its own special code.

Distributed: Writing programs developed in Java that operate over the internet or on a network—web applications or online services—is relatively easy.

High Performance: Multifold speed up has been gained by Java over the years, and some of its advanced features make it really fast at running programs.

Dynamic: Java can change and adapt while a program is running; hence, it is capable of loading new code and working with it right away.

Extensive Standard Library: Java has a vast quantity of tools and libraries that you can use for reading files, accessing the internet and implementing user interfaces.

Architecture-Neutral: Java code can run on any device. This means that it does not depend upon the hardware used in the device or the operating system installed on it. This happens because Java code is translated into universal format.

Network-Centric: Java is especially good at building programs that work on a network—like ones that communicate over the internet.

Scalable: Whether it is tiny apps or enterprise-level, complex systems, Java can be used for all. Basically, it scales well for different needs.

2. What is JDK and JRE?

Acronyms : **JDK** refers to the **Java Development Kit**, while **JRE** refers to the **Java Runtime Environment**. These are the two main components of the Java programming environment.

A **JDK** is a full Java developing kit; it provides groups of tools for the creation and running of Java applications—it consists of all the necessary components to write, test, and debug Java programs. The JRE or Java Runtime Environment is used for running Java applications with a javac compiler that compiles the Java source code into byte-code. Additional tools are debuggers used to eliminate errors or bugs, with tools and libraries for extra development.

JRE stands for Java Runtime Environment: it is ensured that a JVM running Java bytecode does not contain the necessary set of helper libraries or written Java classes to perform different tasks and thus the JRE does not provide general development and code compiling tools. The JRE includes the JVM (Java Virtual Machine) that runs Java bytecodes with which programs can run on different devices and operating systems, as well as the standard libraries precompiled into Java classes for a set of typical tasks.

3. Differentiate between procedure oriented programming and object oriented programming.

Paradigm:

POP: A procedure-oriented programming paradigm focuses on procedures or routines, usually functions, that operate on data. The key emphasis would therefore be on the sequence of tasks.

OOP: An object-oriented programming paradigm focuses on the objects, which combine data and the methods that operate on them. Here, the key focus is on the way that code is organized around objects and classes.

Data Handling:

POP: The basic approach is to have data independent of functions. The style is usually to pass data between functions as arguments.

OOP: Data along with the functions go inside an object. The data gets manipulated by the methods inside the object that offer a much better control and security.

Code Organization:

POP: A program has been organized around functions or procedures. Functions are called one after another to execute any task.

OOP: A program has been organized around objects and classes. Classes can define how the objects will look along with the functions. It leads to modularity and reusability.

Reusability:

POP: Here, reusability of the code is less usually because functions are usually developed for a particular context and need to be rewritten for other contexts.

OOP: This fosters reusability due to inheritance and polymorphism. Classes can be extended or modified without changing the existing code.

Flexibility and Maintenance:

POP: Since it is not encapsulated, a change in one part of the code may affect other parts of it; hence, maintenance becomes tough.

OOP: This enhances flexibility and maintainability because changes in one class do not affect the other. This makes managing and updating codes easier.

Examples:

POP: Languages like C.

OOP: Programs in languages like Java, C++, and Python are object-oriented.

Outcomes:

Describe and explain the basic concepts of programming and tools associated with Java, its features, development environments like JDK and JRE, and finally, its programming paradigms: procedure-oriented versus object-oriented.

Conclusion: (Conclusion to be based on the outcomes achieved)

I have been able to understand the basics or core concepts in programming and affiliated tools in Java within the Tutorial 1. I have tried to understand first and then explained Java's key features, differentiated between the Java Development Kit and Java Runtime Environment and also Procedure-Oriented Programming with Object-Oriented Programming.

Grade: AA / AB / BB / BC / CC / CD /DD

Signature of faculty in-charge with date

References Books

1. Herbert Schildt; JAVA The Complete Reference; Seventh Edition, Tata McGraw- Hill Publishing Company Limited 2007.
2. Java 7 Programming - Black Book : Kogent Learning Solutions Inc.
3. Sachin Malhotra, Saurabh Chaudhary "Programming in Java", Oxford University Press, 2010

4. Jaime Nino, Frederick A. Hosch, 'An introduction to Programming and Object Oriented Design using Java', Wiley Student Edition.