Introduction to Java

Java is a class-based, object-oriented programming language that is designed to have as few implementation dependencies as possible. It is intended to let application developers write once, and run anywhere (WORA), meaning that compiled Java code can run on all platforms that support Java without the need for recompilation. Java was first released in 1995 and is widely used for developing applications for desktop, web, and mobile devices. Java is known for its simplicity, robustness, and security features, making it a popular choice for enterprise-level applications.

JAVA was developed by James Gosling at **Sun Microsystems** Inc in the year **1995** and later acquired by Oracle Corporation. It is a simple programming language. Java makes writing, compiling, and debugging programming easy. It helps to create reusable code and modular programs. Java is a class-based, object-oriented programming language and is designed to have as few implementation dependencies as possible. A general-purpose programming language made for developers to *write once run anywhere* that is compiled Java code can run on all platforms that support Java. Java applications are compiled to byte code that can run on any Java Virtual Machine. The syntax of Java is similar to c/c++.
History: Java's history is very interesting. It is a programming language created in 1991. James Gosling, Mike Sheridan, and Patrick Naughton, a team of Sun engineers known as the **Green team** initiated the Java language in 1991. **Sun Microsystems** released its first public implementation in 1996 as **Java 1.0**. It provides no-cost -run-times on popular platforms. Java1.0 compiler was re-written in Java by Arthur Van Hoff to strictly comply with its specifications. With the arrival of Java 2, new versions had multiple configurations built for different types of platforms.
In 1997, Sun Microsystems approached the ISO standards body and later formalized Java, but it soon withdrew from the process. At one time, Sun made most of its Java implementations available without charge, despite their proprietary software status. Sun generated revenue from Java through the selling of licenses for specialized products such as the Java Enterprise System.

On November 13, 2006, Sun released much of its Java virtual machine as free, open-source software. On May 8, 2007, Sun finished the process, making all of its JVM's core code available under open-source distribution terms.

The principles for creating java were simple, robust, secured, high-performance, portable, multi-threaded, interpreted, dynamic, etc. In 1995 Java was developed by **James Gosling**, who is known as the Father of Java. Currently, Java is used in mobile devices, internet programming, games, e-business, etc.

Implementation of a Java application program involves a following step. They include:
1. Creating the program
2. Compiling the program
3. Running the program

Remember that, before we begin creating the program, the Java Development Kit (JDK) must be properly installed on our system and also path will be set.

- Creating Program

  We can create a program using Text Editor (Notepad) or IDE (NetBeans)

```
        class Test
        {
                public static void main(String []args)
                {
                        System.out.println("My First Java Program.");
                }
        };
```

File -> Save -> d:\Test.java

- Compiling the program

  To compile the program, we must run the Java compiler (javac), with the name of the source file on "command prompt" like as follows

  If everything is OK, the "javac" compiler creates a file called "Test.class" containing byte code of the program.

- Running the program

  We need to use the Java Interpreter to run a program.

Java programming language is named JAVA. Why?

After the name OAK, the team decided to give it a new name to it and the suggested words were Silk, Jolt, revolutionary, DNA, dynamic, etc. These all names were easy to spell and fun to say, but they all wanted the name to reflect the essence of technology. In accordance with James Gosling, **Java** the among the top names along with **Silk**, and since java was a unique name so most of them preferred it.
Java is the name of an **island** in Indonesia where the first coffee(named java coffee) was produced. And this name was chosen by James Gosling while having coffee near his office. Note that Java is just a name, not an acronym.

Java Terminology

Before learning Java, one must be familiar with these common terms of Java.

1. Java Virtual Machine(JVM): This is generally referred to as JVM. There are three execution phases of a program. They are written, compile and run the program.

- Writing a program is done by a java programmer like you and me.

- The compilation is done by the **JAVAC** compiler which is a primary Java compiler included in the Java development kit (JDK). It takes the Java program as input and generates bytecode as output.
- In the Running phase of a program, **JVM** executes the bytecode generated by the compiler.

Now, we understood that the function of Java Virtual Machine is to execute the bytecode produced by the compiler. Every Operating System has a different JVM but the output they produce after the execution of bytecode is the same across all the operating systems. This is why Java is known as a **platform-independent language.**

2. Bytecode in the **Development Process:** As discussed, the Javac compiler of JDK compiles the java source code into bytecode so that it can be executed by JVM. It is saved as **.class** file by the compiler. To view the bytecode, a disassembler like javap can be used.

3. Java Development Kit(JDK): While we were using the term JDK when we learn about bytecode and JVM. So, as the name suggests, it is a complete Java development kit that includes everything including compiler, Java Runtime Environment (JRE), java debuggers, java docs, etc. For the program to execute in java, we need to install JDK on our computer in order to create, compile and run the java program.

4. Java Runtime Environment (JRE): JDK includes JRE. JRE installation on our computers allows the java program to run, however, we cannot compile it. JRE includes a browser, JVM, applet support, and plugins. For running the java program, a computer needs JRE.

5. Garbage Collector: In Java, programmers can't delete the objects. To delete or recollect that memory JVM has a program called Garbage Collector. Garbage Collectors can recollect the objects that are not referenced. So Java makes the life of a programmer easy by handling memory management. However, programmers should be careful about their code whether they are using objects that have been used for a long time. Because Garbage cannot recover the memory of objects being referenced.

6. ClassPath: The classpath is the file path where the java runtime and Java compiler look for **.class** files to load. By default, JDK provides many libraries. If you want to include external libraries they should be added to the classpath.

Primary/Main Features of Java

1. Platform Independent: Compiler converts source code to bytecode and then the JVM executes the bytecode generated by the compiler. This bytecode can run on any platform be it Windows, Linux, or macOS which means if we compile a program on Windows, then we can run it on Linux and vice versa. Each operating system has a different JVM, but the output produced by all the OS is the same after the execution of the bytecode. That is why we call java a platform-independent language.

2. Object-Oriented Programming Language: Organizing the program in the terms of a collection of objects is a way of object-oriented programming, each of which represents an instance of the class.

The four main concepts of Object-Oriented programming are:

- Abstraction
- Encapsulation
- Inheritance
- Polymorphism

3. Simple: Java is one of the simple languages as it does not have complex features like pointers, operator overloading, multiple inheritances, and Explicit memory allocation.

4. Robust: Java language is robust which means reliable. It is developed in such a way that it puts a lot of effort into checking errors as early as possible, that is why the java compiler is able to detect even those errors that are not easy to detect by another programming language. The main features of java that make it robust are garbage collection, Exception Handling, and memory allocation.

5. Secure: In java, we don’t have pointers, so we cannot access out-of-bound arrays i.e it shows **ArrayIndexOutOfBound Exception** if we try to do so. That’s why several security flaws like stack corruption or buffer overflow are impossible to exploit in Java. Also, java programs run in an environment that is independent of the os(operating system) environment which makes java programs more secure.

6. Distributed: We can create distributed applications using the java programming language. Remote Method Invocation and Enterprise Java Beans are used for creating distributed applications in java. The java programs can be easily distributed on one or more systems that are connected to each other through an internet connection.

7. Multithreading: Java supports multithreading. It is a Java feature that allows concurrent execution of two or more parts of a program for maximum utilization of the CPU.

8. Portable: As we know, java code written on one machine can be run on another machine. The platform-independent feature of java in which its platform-independent bytecode can be taken to any platform for execution makes java portable.

9. High Performance: Java architecture is defined in such a way that it reduces overhead during the runtime and at some times java uses Just In Time (JIT) compiler where the compiler compiles code on-demand basics where it only compiles those methods that are called making applications to execute faster.

10. Dynamic flexibility: Java being completely object-oriented gives us the flexibility to add classes, new methods to existing classes, and even create new classes through sub-classes. Java even supports functions written in other languages such as C, C++ which are referred to as native methods.

11. Sandbox Execution: Java programs run in a separate space that allows user to execute their applications without affecting the underlying system with help of a bytecode verifier. Bytecode verifier also provides additional security as its role is to check the code for any violation of access.

12. Write Once Run Anywhere: As discussed above java application generates a ‘.class’ file that corresponds to our applications(program) but contains code in binary format. It provides ease t architecture-neutral ease as bytecode is not dependent on any machine architecture. It is the primary reason java is used in the enterprising IT industry globally worldwide.

13. Power of compilation and interpretation: Most languages are designed with the purpose of either they are compiled language or they are interpreted language. But java integrates arising enormous power as Java compiler compiles the source code to bytecode and JVM executes this bytecode to machine OS-dependent executable code.

class : class keyword is used to declare classes in Java
public : It is an access specifier. Public means this function is visible to all.
static : static is again a keyword used to make a function static. To execute a static function you do not have to create an Object of the class. The main() method here is called by JVM, without creating any object for class.
void : It is the return type, meaning this function will not return anything.
main : main() method is the most important method in a Java program. This is the method which is executed, hence all the logic must be inside the main() method. If a java class is not having a main() method, it causes compilation error.
String[] args : This is used to signify that the user may opt to enter parameters to the Java Program at command line. We can use both String[] args or String args[]. Java compiler would accept both forms.
System.out.println : This is used to print anything on the console like “printf” in C language.

Example

- Java

```
// Importing classes from packages
import java.io.*;

// Main class
public class GFG {

    // Main driver method
    public static void main(String[] args)
    {

        // Print statement
        System.out.println("Welcome to GeeksforGeeks");
    }
}
```

Output

```
Welcome to GeeksforGeeks
```

- Java

```java
// Q: Write a Java program to calculate the sum and product of two given number.


class Sample {

    public static void main(String args[])

    {

        int a,b,sum=0,product=0;

        a=10;

        b=20;

        sum=a+b;

        product=a*b;

        System.out.println("Sum="+sum);

        System.out.println("Product="+product);

    }

};
```

- Java

```java
// Q: Write a Java program to accept two numbers from the user and calculate sum and product.


import java.util.*;

   class Sample {

    public static void main(String args[])

    {

        int a,b,sum=0,product=0;

        Scanner scan=new Scanner(System.in);

        System.out.println("Enter the First No:");

        a=scan.nextInt();
        System.out.println("Enter the Second No:");

        b=scan.nextInt();

        sum=a+b;

        product=a*b;

        System.out.println("Sum="+sum);

        System.out.println("Product="+product);

    }

}
```

- Java

```
// Q: Write a Java program to accept two number from the user and calculate the average.


import java.util.*;

public class Sample {

    public static void main(String args[])

    {

       double a,b,sum=0,avg=0;

        Scanner scan=new Scanner(System.in);

        System.out.println("Enter the First No:");

        a=scan.nextDouble();

        System.out.println("Enter the Second No:")

        b=scan.nextDouble();

        sum=a+b;

        avg=sum/2;

        System.out.println("Average="+avg);

    }

}
```

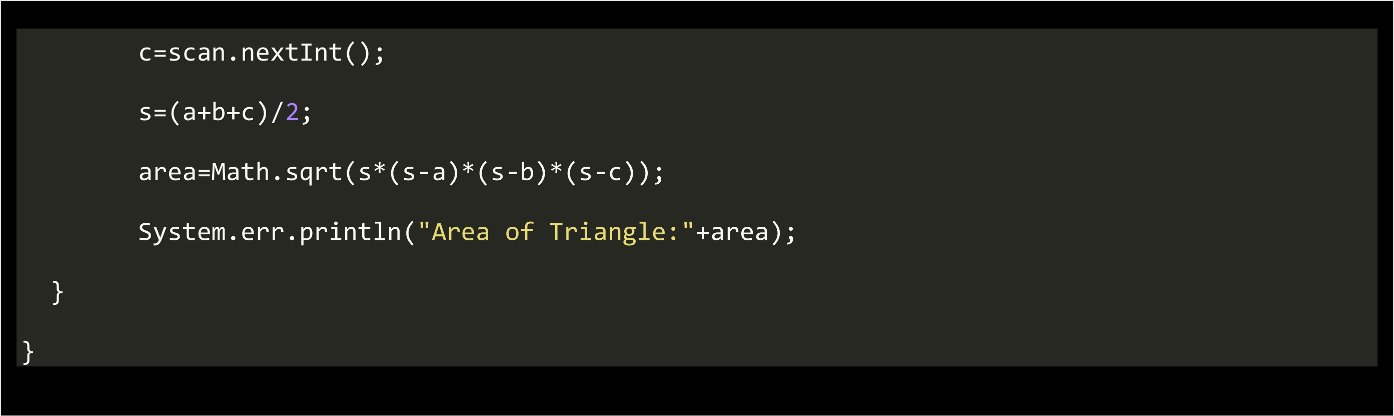
- Java

```
// Q: Write a java program to convert fahrenheit to celsius and celsius to Fahrenheit using formula
```

```java
import java.util.Scanner;

public class Fahrenheit_Celsius

{

    public static void main(String[] args)

    {

        double c, f;

        Scanner s = new Scanner(System.in);

        System.out.print("Enter temperature in Fahrenheit:");

        f = s.nextDouble();

        c = (5.0/9.0)*(f-32);

        System.out.println("Temperature in Celsius:"+c);

    }

}
```

- Java

```java
// Q: Write a Java program to calculate the area of triangle using three sides.


import java.lang.*;

import java.util.Scanner;

public class Area

{

    public static void main(String[] args)

    {


        int a,b,c;

        double s,area;

        Scanner scan = new Scanner(System.in);

        System.out.print("Enter the three sides:");

        a=scan.nextInt();

        b=scan.nextInt();
```



```
        c=scan.nextInt();

        s=(a+b+c)/2;

        area=Math.sqrt(s*(s-a)*(s-b)*(s-c));

        System.err.println("Area of Triangle:"+area);

  }

}
```

Explanation:

1. Comments: Comments are used for explaining code and are used in a similar manner in Java or C or C++. Compilers ignore the comment entries and do not execute them. Comments can be of a single line or multiple lines.

Single line Comments:

Syntax:

```
// Single line comment
```

Multi-line **comments:**

Syntax:

```
/* Multi line comments*/
```

2. import java.io.*: This means all the classes of the io package can be imported. Java io package provides a set of input and output streams for reading and writing data to files or other input or output sources.

3. class: The class contains the data and methods to be used in the program. Methods define the behavior of the class. Class **GFG** has only one method Main in JAVA.

4. static void main(): static keyword tells us that this method is accessible without instantiating the class.

5. void: keywords tell that this method will not return anything. The main**()** method is the entry point of our application.

6. System.in: This is the **standard input stream** that is used to read characters from the keyboard or any other standard input device.

7. System.out: This is the **standard output stream** that is used to produce the result of a program on an output device like the computer screen.

8. println(): This method in Java is also used to display text on the console. It prints the text on the console and the cursor moves to the start of the next line at the console. The next printing takes place from the next line.

9. String []args: This is the argument passed to the main function which is an array of strings with the array name args. One can choose their own flexible name but this name is used by many developers.

Everything in java is represented in Class as an object including the main function.

IMPORTANT POINTS OF JAVA:

Here are some important points about Java:

1. Java is a high-level, object-oriented programming language that was first released in 1995.
2. Java is platform-independent, which means that code written in Java can run on any platform that has a Java Virtual Machine (JVM) installed.
3. Java code is compiled into bytecode, which can then be executed by the JVM.
4. Java is known for its “write once, run anywhere” philosophy, which makes it a popular choice for cross-platform development.
5. Java provides automatic memory management through garbage collection, which makes it easier to write and maintain code.
6. Java has a vast standard library that provides a wide range of tools for common programming tasks.
7. Java is widely used in enterprise applications, web development, and Android app development.
8. Java is a strongly typed language, which means that every variable and expression has a specific type that must be declared before use.
9. Java has a robust exception-handling mechanism that makes it easier to handle errors and unexpected behavior in code.
10. Java supports multithreading, which makes it possible to write programs that can perform multiple tasks simultaneously.

Advantages of Java:

1. Platform independent: Java code can run on any platform that has a Java Virtual Machine (JVM) installed, which means that applications can be written once and run on any device.
2. Object-Oriented: Java is an object-oriented programming language, which means that it follows the principles of encapsulation, inheritance, and polymorphism.
3. Security: Java has built-in security features that make it a secure platform for developing applications, such as automatic memory management and type checking.
4. Large community: Java has a large and active community of developers, which means that there is a lot of support available for learning and using the language.
5. Enterprise-level applications: Java is widely used for developing enterprise-level applications, such as web applications, e-commerce systems, and database systems.

Disadvantages of Java:

1. Performance: Java can be slower compared to other programming languages, such as C++, due to its use of a virtual machine and automatic memory management.
2. Memory management: Java's automatic memory management can lead to slower performance and increased memory usage, which can be a drawback for some applications.