JAVA PROGRAMMING LANGUAGE

Agenda

1. What is JAVA?

2. Features of Java

3. Hello World Program

4. Brief Concept JDK , JRE and JVM

5. Java OOP Concepts

What is Java?

- Java is a **Programming language** and a **Platform**
- Any hardware or software environment in which a program runs, known as a platform. Since Java has its own Runtime Environment (JRE) and API, it is called **platform**.
- **Java Version**

1. JDK Alpha and Beta (1995) JDK 1.0 (23rd Jan, 1996)
2. JDK 1.1 (19th Feb, 1997) J2SE 1.2 (8th Dec, 1998)
3. J2SE 1.3 (8th May, 2000) J2SE 1.4 (6th Feb, 2002)
4. J2SE 5.0 (30th Sep, 2004) Java SE 6 (11th Dec, 2006)
5. **Java SE 7 (28th July, 2011)**

What it is Used?

1. Desktop Applications such as acrobat reader, media player, antivirus etc.
2. Web Applications such as irctc.co.in..etc.
3. Enterprise Applications such as banking applications.
4. Mobile
5. Embedded System
6. Smart Card
7. Robotics
8. Games etc.

Types of Java Application?

- There are mainly 4 type of applications that can be created using java:
- 1) **Standalone Application**

  It is also known as desktop application or window-based application. An application that we need to install on every machine such as media player, antivirus etc. AWT and Swing are used in java for creating standalone applications.

- 2) **Web Application**

  An application that runs on the server side and creates dynamic page, is called web application. Currently, servlet, jsp, struts, jsf etc. technologies are used for creating web applications in java.

Types of Java Application

- 3) **Enterprise Application**

  An application that is distributed in nature, such as banking applications etc. It has the advantage of high level security, load balancing and clustering. In java, EJB is used for creating enterprise applications.

- 4) **Mobile Application**

  An application that is created for mobile devices. Currently Android and Java ME are used for creating mobile applications.

Brief History of JAVA

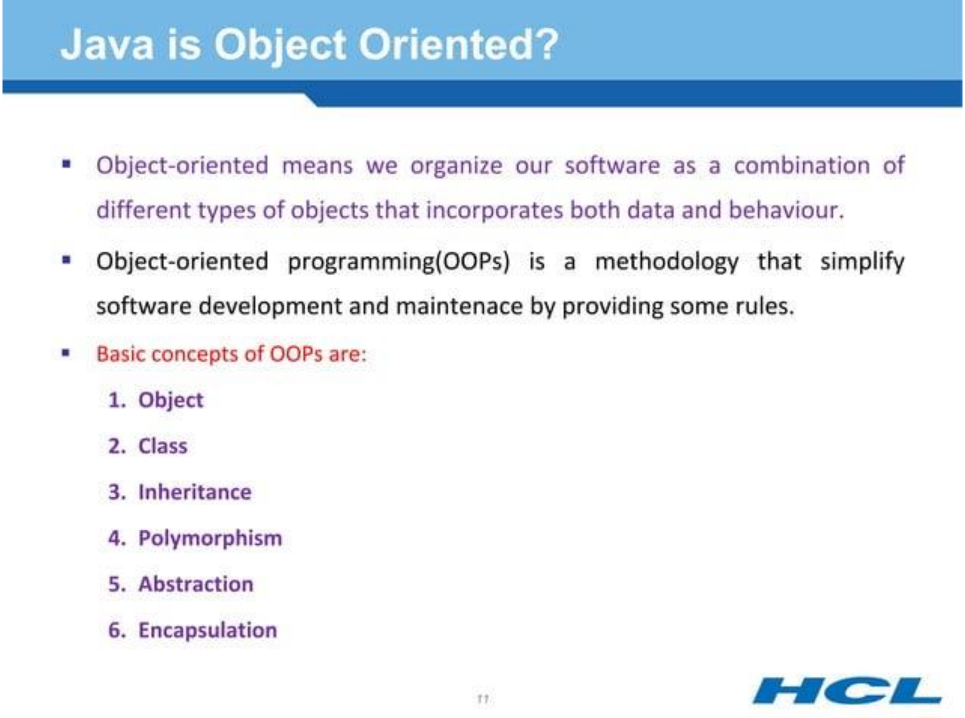
- James Gosling, Mike Sheridan, and Patrick Naughton initiated the Java language project in June 1991.
- originally designed for small, embedded systems in electronic appliances like set-top boxes.
- initially called Oak and was developed as a part of the Green project
- In 1995, Oak was renamed as "Java". Java is just a name not an acronym.
- originally developed by James Gosling at Sun Microsystems(which is now a subsidiary of Oracle Corporation) and released in 1995.
- **JDK 1.0 released in(January 23, 1996).**

Features of Java

- Simple
- Object-Oriented
- Platform Independent
- Secured
- Robust
- Architecture Neutral
- Portable
- High Performance
- Distributed
- Multi-threaded

Java is Simple??

- Java is simple in the sense that:
- syntax is based on C++ (so easier for programmers to learn it after C++).
- removed many confusing and/or rarely-used features e.g., explicit pointers, operator overloading etc.
- No need to remove unreferenced objects because there is Automatic Garbage Collection in java.



Java is Object Oriented?

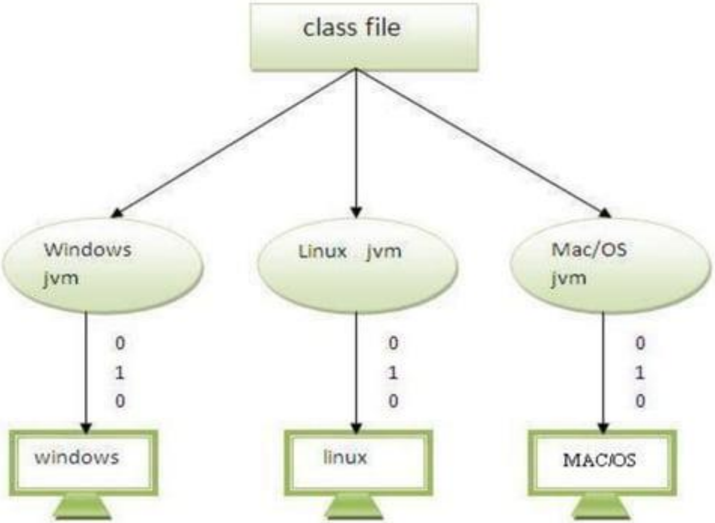
- Object-oriented means we organize our software as a combination of different types of objects that incorporates both data and behaviour.
- Object-oriented programming(OOPs) is a methodology that simplify software development and maintenace by providing some rules.
- Basic concepts of OOPs are:

1. **Object**
2. **Class**
3. **Inheritance**
4. **Polymorphism**
5. **Abstraction**
6. **Encapsulation**

Java is Platform Independent?

- A platform is the hardware or software environment in which a program runs. There are two types of platforms software-based and hardware-based. Java provides software-based platform. The Java platform differs from most other platforms in the sense that it's a software-based platform that runs on top of other hardware-based platforms. It has two components:
- **Runtime Environment & API(Application Programming Interface)**
- Java code can be run on multiple platforms e.g. Windows,Linux,Sun Solaris,Mac/OS etc.
- Java code is compiled by the compiler and converted into bytecode.
- This **bytecode** is a platform independent code because it can be run on multiple platforms i.e. Write Once and Run Anywhere(WORA).

Platform Independent



Java is Secure

- **Java is Secured because:**
- No explicit pointer
- Programs run inside virtual machine sandbox.
- Class loader- adds security by separating the package for the classes of the local file system from those that are imported from network sources.
- Byte code Verifier- checks the code fragments for illegal code that can violate accesss right to objects.
- Security Manager- determines what resources a class can access such as reading and writing to the local disk.

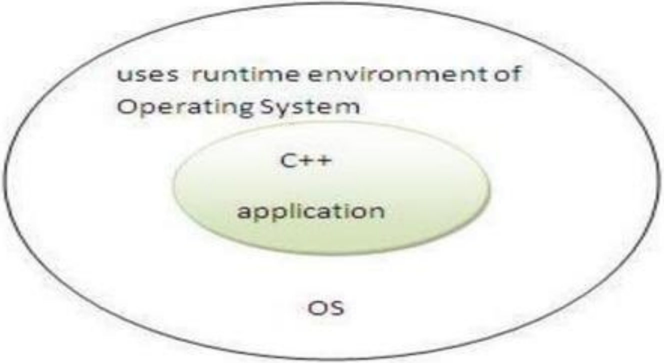
Some security can also be provided by through SSL, JAAS,cryptography etc.

Java is Robust

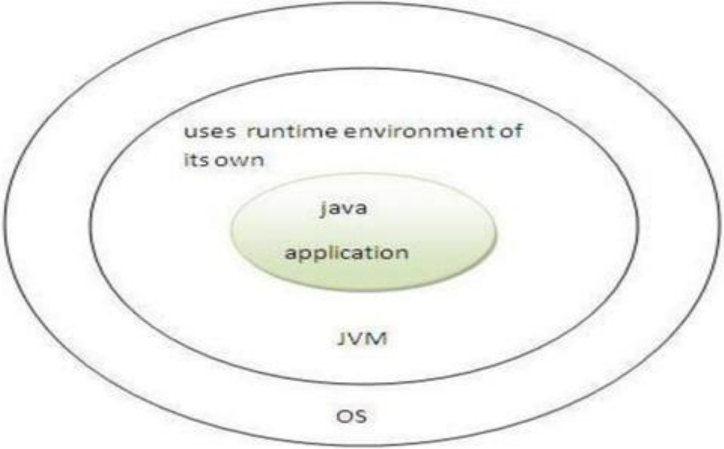
Java is Robust because -

- Robust simply means strong. Java uses strong memory management. There are lack of pointers that avoids security problem. There is automatic garbage collection in java. There is exception handling and type checking mechanism in java. All these points makes java robust.
- Architecture-neutral
- There is no implementation dependent features e.g. size of primitive types is set.

C++ Program Execution



Java Program Execution



Multi-Threaded

- **Multi-threaded**

  A thread is like a separate program, executing concurrently. We can write Java programs that deal with many tasks at once by defining multiple threads. The main advantage of multi-threading is that it shares the same memory. Threads are important for multi-media, Web applications etc.

Hello Word Java Program!!

- Requirement for Hello Java Example

For executing any java program, you need to

- create the java program.
- install the JDK if you don't have installed it You may also use the eclipse different version.
- set path of the bin directory under jdk. In Case of Eclipse you may also do the same.
- compile and execute the program.

First Java Application

- File Name=Class Name
- Syntax:-

```
class ClassName
{
        public static void main(String[] args)
        {
                //Lines of Code1
        }
}
```

- Example:-

```
class MyFirstProg                                        (MyFirstProg.java)
{
        public static void main(String[] args)
        {
                System.out.println(“This is my First Java Program”);
        }
}
```

Hello Word Java Program!!

Creating hello java example

Let's create the hello java program:

```
class Simple{
   public static void main(String args[]){
    System.out.println("Hello Java")
   }
}
```

save this file as Simple.java

Understanding First Java Program

- **class** is used to declare a class in java.
- **public** is an access modifier which represents visibility, it means it is visible to all.
- **static** is a keyword, if we declare any method as static, it is known as static method. The core advantage of static method is that there is no need to create object to invoke the static method. The main method is executed by the JVM, so it doesn't require to create object to invoke the main method. So it saves memory.
- **void** is the return type of the method, it means it doesn't return any value.

Why Standard Syntax for Java main() Method?

- public static void main(String[] args)
  - ❖ public -> Access Specifier.
  - ❖ static -> Access Modifier.
  - ❖ void -> Return Type of Method.
  - ❖ main -> Name of the Method. (Starting Point of Execution).
  - ❖ String[] args -> Parameter of main method and accepts command line arguments during program execution.