Introduction to Java

History of Java

- James Gosling initiated Java in 1991 for use in one of his project at Sun microsystems
- Named as Oak, then called as Green, Later it was finally named as Java
- Released in 1995

Features of Java

1. Simple
2. Object-Oriented
3. Platform independent
4. Secured
5. Robust
6. Architecture neutral
7. Portable
8. Dynamic
9. Interpreted
10. High Performance
11. Multithreaded
12. Distributed

Feature - Simple

- 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

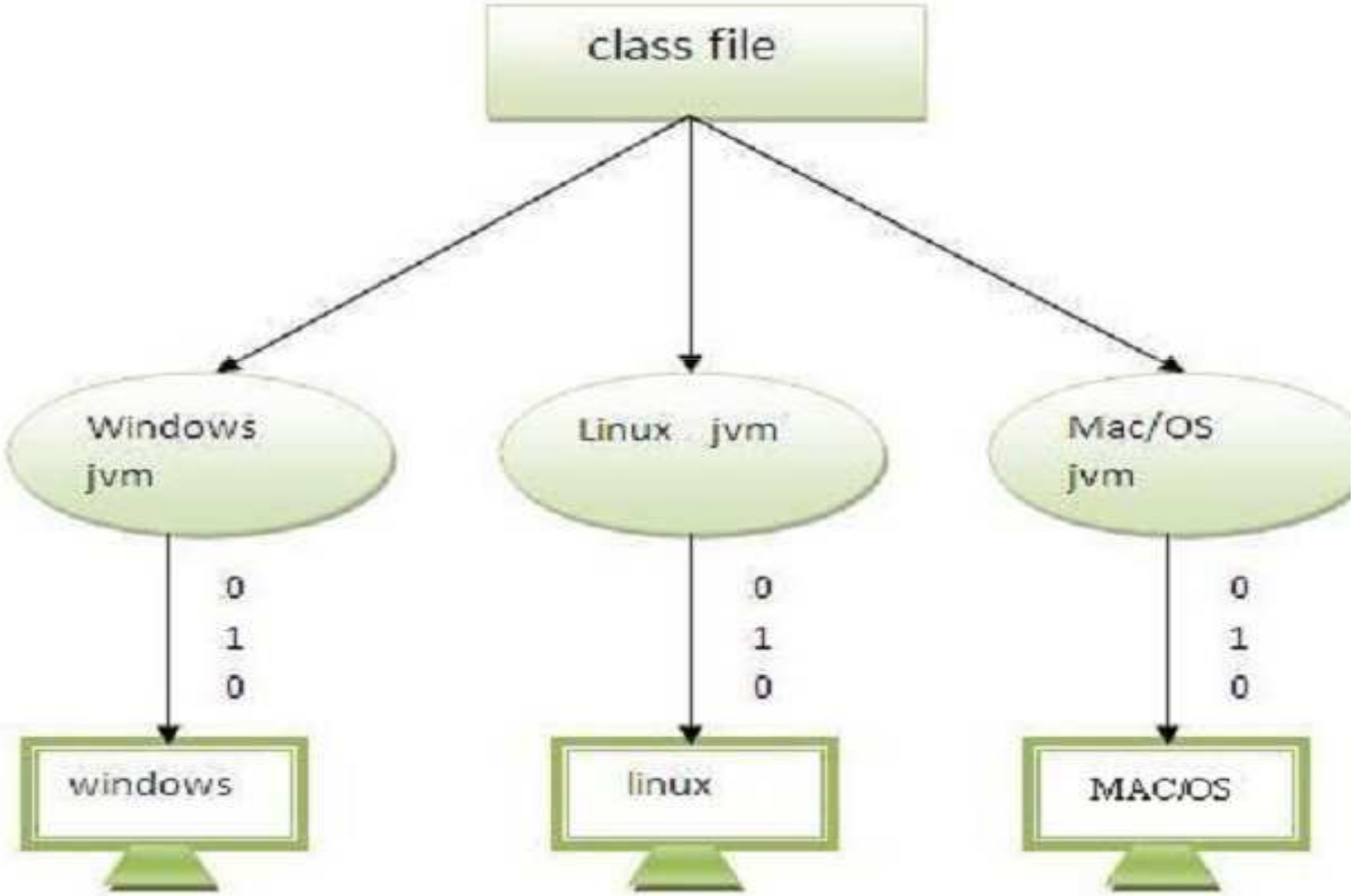
Feature – Object oriented

- Object-oriented means - Any 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 maintenance** by providing some rules.

Feature – Platform Independent

- **Java Compiler** – Compiles the Java code into **Bytecode**
- **JVM is Java Virtual Machine** -- Runs/ Interprets/ translates Bytecode into **Native Machine Code**
- JVM is a virtual platform that resides on your RAM
  - Its component, **Class loader** loads the .class file into the RAM
  - The **Byte code Verifier** component in JVM checks if there are any access restriction violations in your code. (This is one of the principle reasons why java is secure)
  - Next, the **Execution Engine** component converts the Bytecode into executable machine code
- This bytecode is a platform independent code because it can be run on multiple platforms i.e. Write Once and Run Anywhere(WORA).

Contd…


class file
Windows jvm
Linux jvm
Mac/OS jvm
0
1
0
0
1
0
0
1
0
windows
linux
MAC/OS

Feature - Security

- Pointers are dangerous
  - lead to memory leaks, memory corruption, invalid memory access, e.g. from uninitialized and improperly initialized variables, indexing out of bounds, and many bugs due to pointer arithmetic.
- Java is secured because:
  - No explicit pointer
  - Programs run inside virtual machine sandbox.
- Classloader- adds security by separating the package for the classes of the local file system from those that are imported from network sources.
- BytecodeVerifier- checks the code fragments for illegal code that can violate access right to objects.
- Security Manager- determines what resources a class can access such as reading and writing to the local disk.

Feature - Robust

- 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

Feature - Portable

- We may carry the java bytecode to any platform.

Feature - Distributed

- We can create distributed applications in java. RMI and EJB are used for creating distributed applications.
- We may access files by calling the methods from any machine on the internet.

Feature - Multithreaded

- 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.

Sample Program

```
/*First Program */
class Example{
public static void main(String args[ ])
{
   System.out.println(“Hello World”);
}
```

} **System** is a class in the **java**.lang package.

out is a static member of the **System** class, and is an instance of **java**.io.PrintStream .

println is a method of **java**.io.PrintStream .

Compilation & Execution

- C:/> javac Example.java
- C:/> java Example
- Output:

  Hello World