**Day 2**

Features of Java continues.......

**3. Platform Independent:** In Java, the source code is compiled to bytecode and this bytecode can be used in any operating system’s JVM without any implementation. So, only once we need to write the programs and compile; this generated bytecode can be exported anywhere independent of the platform.

**4.** **Secured:** Java is considered as secured because,

1. In Java, no explicit pointers are used. While using pointers concept, the address of the memory location is being fetched. Thus, there will be lack of security. But in Java since we are not using this pointers concept, it is more secured.
2. Since, Java programs are runed inside the JVM, its security is more.

**5. Robust:** java programming language is robust which means strong because,

1. In Java automatic garbage collection is maintained. So that the memory space is more since it automatically clears the unreferenced memory locations. So, it has a strong memory management.
2. Since pointers are not used in Java, it is secure.
3. Java handles unexpected errors through Exception handling mechanism. That is, we can manage the compile time errors, but run time errors are managed by Java through Exception handling.

**6. Architectural Neutral:** In C programming, 32-bit architecture occupies 2 bytes of memory for int data type and 64-bit architecture occupies 4 bytes of memory. But in Java, for both 32-bit as well as 64-bit architecture we can use 2 byte or 4 bytes of memory as we like.

That’s is, Java is architecture neutral because there are no implementation dependent features, for example, the size of primitive types is fixed.

**7. Portable:** Java programs are portable, which means that the same bytecode program can run on any computer system that has a Java interpreter. Also, a source program can be compiled into bytecodes on any computer that has a Java compiler.

**8. High Performance:** Comparing to C and C++ language, Java program has high performance, since its byte code is very close to machine language. So, Java has high performance.

**9. Distributed:** Java is distributed because it encourages users to create distributed applications. In Java, we can split a program into many parts and store these parts on different computers. A Java programmer sitting on a machine can access another program running on the other machine.

**10. Multithreaded:** In Java, small programs are called threads. These threads can be partially executed and they use a common memory area. So, memory issues will not be there. Eg. Web applications. In Java, Multithreading refers to a process of executing two or more threads simultaneously for maximum utilization of the CPU**.**

**11. Dynamic:** In Java, based on the program flow, classes are loaded dynamically. Thus, on demand classes are loaded in memory.

**12. Interpreted:** In Java, to convert the bytecode to machine language, an interpreter called Java interpreter is used by the compiler. Interpreter in Java is a computer program that converts high-level program statement into Assembly Level Language.

**Variable in Java**

In Java, variables are used as a container. The values in variables can be changed.

int a = 2; // here a is the variable name given to the memory location.

**Types of Variables:**

There are three types of variables in java:

1. Local Variables
2. Instance variables
3. Static Variables

**Local variables:** Local variables are declared inside the method. The scope of the local variables is only inside the method not outside.

**Instance Variables:** Instance Variables are declared inside the class but outside the method. For these types of variables memory allocations will be dynamic.

**Static Variables:** These variables are used with the keyword ‘Static’. The memory allocation for these types of variables is done only once.

Class A

{

Static int a=10; // static variable

Void method()

{

int b=12; // local variable

}

public static void main(String args[])

{

int c =30; // instance variable

}

}