Java is Platform Independent

-

-platform is the environment in which a program runs.

-Now being platform independent means that an application developed and compiled over one platform can be executed over any other platform without any change in the code

-Whenever we compile a java program, the compiler never generates machine code. Rather it generates a machine independent code called the “byte code” .

-this byte code is not directly understandable by the platform(platform is the combination of OS+CPU)

-So another special layer of software is required to convert these byte code instructions to machine dependent form

-This special layer is the JVM, that converts the byte code to machine code

-Thus any such platform in which a JVM is available can be used to execute a java application

Automatic Memory Management -

-In languages like C and C++ any dynamic memory which the programmer allocates using malloc() or new has to be deallocated by himself using free() or delete

-but java uses a runtime automatic garbage collection feature where the JVM itself deallocates any dynamic memory which our program allocated.

Secure -

-when it comes to Security java is always

the first choice. It enables us to develop virus free, temper free System.

-Java is a more secure Language as Compared to C/C++ because -

a. It does not allow programmer to Explicitly create pointers

b. Java program always runs in Java runtime Environment with almost null interaction with system OS, hence it is more secure.

Robust -

-Java has very Strict rules which every Progam must Compulsory follow and if these rules are violated then JVM kills/ terminates the code by generating “Exception”.

Object Oriented -

-Java Supports all important concept of OOPs like Inheritance, Encapsulation, Polymorphism, Abstraction

Multithreaded -

-Multithreaded means Concurrent execution.

-in simple terms we can execute more than one part of the same program parallely/Simultaneously

```
PSVM()
{
clrscr() ;
factorial(5) ;
Prime(8) ;
evenodd(4) ;
```

}

- In the above code previous sample code all 4 functions are independent of each other but still they will run sequentially i.e., one after the other

-This can be improved in java by using Multithreading features so that all of these functions can run together

* Benefits- Reduced Execution time, full Utilisation of CPU

JDK (v/s) JRE (v/s) JVM

1. JVM(Java Virtual Machine) -

-JVM is an abstract Machine that can execute precompiled Java Programs.

-In Simple terms it is the code execution

Component of Java .

-It is designed for each platform(OS+CPU) Supported by java and this means that every platform will have a different Version of JVM

*Question - Why JVM is Called a virtual Machine?

-JVM is called Virtual Machine because it is a software layer but it behaves as if it is a complete Machine(platform).

-That is all the task which are done by a machine while running a program in other language like c, are actually done by JVM in java.

For Example :-

Starting the Execution by Calling main() method

Allocating memory for the Program

Cleaning up memory

etc…

-What JVM contains?

a. Interpreter

b. Garbage Collector

*Question - Are Java Compiler and Interpreter same?

-ans is NO, Not at all

-The Java Compiler Converts Source Code to byte code and is not a part of JVM , rather it comes with JDK.

-The interpreter lives inside the JVM and Converts byte code to Machine Understandable code.

2. JRE(Java Runtime Environment) -

- It contains a JVM along with java classes/packages and set of runtime libraries .

-So the JVM, while running a Java Program uses the classes and other libraries supplied by JRE.

-If we do not want to write a java program and we just want to run it then we only need JRE.

3. JDK(Java Development Kit) -

-JDK is a bundle of software that we can use to develop Java based applications.

-It includes the JRE, Set of Libraries class, Java Compiler, jar and addtoional tools needed while Developing a Java application

Q) can I compile a java application if I have JRE?

ans: No

JRE can only be used to run a Java application. It doesnot Contains the javac tool which is used for Compilation

Q) Which Component is used to Compile, Debug, and execute java program?

a) JVM b) JDK c) JIT d) JRE

ans: B

Q) Which Component is used to convert bytecode to machine specific code?

a) JVM b) JDK c) JIT d) JRE

ans: A

Q) Which Component is used to provide a platform to run a java program?

a) JVM b) JDK c) JIT d) JRE

ans: D

Q) Why main() is public?

ans: “public” is an access modifier and in OOP any method or variable which is declared public can be accessible from outside of the class.

since main() is public so JVM can easily access and execute it.

Q) Why is main() static?

ans: A static method can be called called without object, simply using class name.

In System.out.println(“message”)—>

-System is a predefined class

-out is an object reference(not object)

-println() is a method

Q)will a java program Compile without main() ?

ans: yes, Because main() is not needed for Compilation but is used for execution of the code. so we can Compile a program without main() but we cannot run it.

DataType

<—

—>

Type Conversion

Whenever the compiler encounters a statement where the value on right side of assignent is different than the variable on left side, then the compiler tries to convert RHS to LHS and this automatic conversion is done by compiler.

-Type Conversion is of two type

1. Implicit Conversion(Automatically done by compiler)

-for Implicit conversion there are 2 condition must be true

rule 1: the value must be Compatible/ convertible.

eg: int x = ‘A’ ; —> correct

int x = true ; —> not correct

rule 2: the value on RHS of assignment must be smaller than variable on LHS.

Smaller means the range of variable's data type must be a smaller than other variables range, NOT THE SIZE.

for example, a short data type variable has a range of -32768 to 32767 which is smaller of the range of int variable whose range is -2147483648 to 2147483647, so "short" is smaller than int .

Type conversion in Expression:

byte a = 10 ;

byte b = 20 ;

byte c = a+b ;

this will give error—> possible loss of precision

because both a and b are byte, java converts them to int

solution for the above problem

c = (byte)(a+b) ; —> not reliable as byte is < int so there may be rotation of value

or

byte a = 10 ;

byte b = 20 ;

int c = a+b ; —> More Reliable

* Necessary Condition for Automatic type Conversion in Java—> the Destination type is larger than than source type.

Q) What is the error in this code?

byte a = 50 ;

a = a*40 ;

sol: * operator has Converted a*40 into int, which cannot be Converted to byte

without casting

Q) If an expression Contains double, int, float, long, then expression will promoted into which of these data types

a) long

b) int

c) double

d) float

ans: double