

## Assignment Questions 1

### Q1.What is the difference between Compiler and Interpreter

**ans:- Compiler** -> It is a software which takes sourcecode(HLL) as the input and generates MLL code as the output

To convert the HLL code to MLL code compiler will scan the HLL code only once.

**Interpreter** =>It is a software which takes sourcecode(HLL) as the input and generates MLL code as the output

To convert the HLL code to MLL code interpreter will scan the HLL code multiple times(depends on the instructions).

### Q2.What is the difference between JDK, JRE, and JVM?

**ans:-** => java program to run we need to set up the environment in our machines

=> To set up the environment we need to install JDK software to our machines.

=> JDK stands for Java Development kit, it provides libraries and the required files to run our java programs

=> JDK :: JRE + JVM

=> JRE :: Java RuntimeEnvironment, It provides suitable environment to run our java program.

=> JVM :: Java Virtual Machine, It is responsible to run our java programs on the basis of MultiThreading.

Program Execution

=====

=> java programs will be first compiled, to compile java program we need java compiler and java compiler will be installed when we install jdk software.

=> jdk software installation location :: c:\programfiles\jdk....

=> All the commands required for java developer to run his program would be present inside <java\_home>\bin folder

=> bin folder

|=> javac

|=> java

|=> javap

|=> javadoc

|=> jar

=> javaprogram would be first compiled(javac filename.java)

=> If the compilation is succesfull, it would generate .class file

=> These .class files will be used by jvm during the execution

=> .class file generated will have instructions in bytecode(neither HLL nor MLL).

=> bytecodes will be taken by JVM and it will be loaded inside JRE, then the execution begins.

Java programs => java compilation + Execution

(javac) (java-> JIT)

### **Q3.How many types of memory areas are allocated by JVM?**

**ans:-**

JVM has five memory locations namely –

Heap – Runtime storage allocation for objects (reference types).

Stack – Storage for local variables and partial results. A stack contains frames and allocates one for each thread. Once a thread gets completed, this frame also gets destroyed. It also plays roles in method invocation and returns.

PC Registers – Program Counter Registers contains the address of an instruction that JVM is currently executing.

Execution Engine – It has a virtual processor, interpreter to interpret bytecode instructions one by one and a JIT, just in time compiler.

Native method stacks – It contains all the native methods used by the application.

#### **Q4.What is JIT compiler?**

**ans:-**

The Just-In-Time (JIT) compiler is a component of the runtime environment that improves the performance of Java™ applications by compiling bytecodes to native machine code at run time.

#### **Q5.What are the various access specifiers in Java?**

**ans->** In java, there are four types of access specifiers and the name of these access specifiers are given below:

public access specifiers

protected access specifiers

default access specifiers

private access specifiers

#### **Q6.What is a compiler in Java?**

**ans:-** It is a software which takes sourcecode(HLL) as the input and generates MLL code as the output

To convert the HLL code to MLL code compiler will scan the HLL code only once.

#### **Q7.Explain the types of variables in Java?**

**ans:-**A variable is assigned with a data type. Variable is a name of memory location. There are three types of variables in java: local, instance and static. There are two types of data types in Java: primitive and non-primitive.

#### **Q8.What are the Datatypes in Java?**

**ans:-** 1. Number

a. Integral type byte,short,int,long

- b. Floating type float,double
- 2. Character
  - a. char
- 3. Boolean type
  - a. boolean
- 4. Images,audio,video
  - a. LOB's(Large Object)

Q9.What are the identifiers in java?

ans-> identifier is a name in java program.

identifier can be classname,methodname,variablename,labelname.

eg::

class Demo

{

public static void main(String[] args)

{

int x = 10;

}

}

Demo: classname

main: methodName

String: className

args : variablename

x : variableName

### **Q10.Explain the architecture of JVM**

**ans:-** JVM preference while loading the .class file

1. Search in the current working directory from where the program would start.
2. Search in the rt.jar(inbuilt jar with all the .class files) file also with the specified packages
3. search in the ext folder of jre environment.
4. search in the classpath supplied jar file