Assignment - 1

1) Explain role of JVM.

- JVM stand for Java Virtual machine.
- JVM take care of creating a class file related to your program and you can use same class file in different operating system.
- JVM acts as a interface between your operating system and your program code.
- It also provides security related to features
- JVM is responsible for calling main method and running your program.
- It is also responsible for calling garbage collector for clean up operations.
- Hence, this are the few reasons and role of Java Virtual machine (JVM)

2) Discuss public, private and protected access modifiers.

- Public: The access level of public modifier is everywhere. It can be accessed from within the class, outside the class, within the package and outside the package.
  It has the widest scope among all other modifiers.

Date:

- **Private :** The access level of private modifier is only within the class. It cannot be accessed from outside the class.
  If you make any class constructor private, you cannot create the instance of that class from outside the class.

- **Proctected :** The access level of a protected modifier is within the package and outside the package through child class.
  If you do not make the child class, it cannot be accessed from outside the package.
  The protected access modifier can be applied on the data member, method and constructor.
  It can't be applied on the class.

3) Explain Types of Operators in JAVA.

- The various types of operator in java are given (Explained) below:

1) **Arithmetic Operators :**

- They are used to perform simple arithmetic operations on primitive data types.
  - Multiplication (*)
  - Division (/)
  - Modulo (%)
  - Addition (+)
  - Substraction (-)

2) Assignment Operator :

- It is used to assign any value to any variable
- The associativity of this operator is right to left.
  - +=
  - -=
  - *=
  - /=
  - %=

3) Relational Operator :

- These operators are used to check for relations like equality, greater than, less than.
- It given output as a Boolean
  - Equal to (==)
  - Not Equal to (!=)
  - Less than (<)
  - Less than equal to (<=)
  - greater than (>)
  - greater than equal to (>=)

4) Logical Operator :

- These operators are used to perform "Logical AND" and "Logical OR" operation.
  - (&&) Logical AND
  - (||) Logical OR

5) Unary Operator :

- It needs only one operator and used to increment, decrement or negate a value.

- Unary minus (-)
- Unary plus (+)
- Increment operator (++)
- Decrement operator (--)
- Logical not operator (!)

6) Ternary operator :

- It is a shorthand version of if-else statement. It has three operands and hence the name ternary.

Syntax:

```
condition ? if true : if false;
```

7) Bitwise Operator :

These operators are used to perform manipulation of individual bits of a number.

- Bitwise AND (&)
- Bitwise OR (|)
- Bitwise XOR (^)
- Bitwise complement (~)

8) Shift Operator :

- This operators are used to shift the bits of a number left or right thereby multiplying or dividing the number by two respectively.
- left shift (<<)
- signed right shift (>>)
- unsigned right shift (>>>)

4) Explain features of Java.

- A few notable features of Java are explained below:

- Object Oriented :
  In java, everything is an object. Java can be easily extended since it is based on the object model.

- Platform Independent :
  Unlike many other programming languages including C and C++, when java is compiled, it is not compiled into platform specific machine, rather into platform-independent byte code.
  This byte code is distributed over the web and interpreted by the virtual Machine (JVM) on whichever platform it is being run on.

- Simple :
  Java is designed to be easy to learn. If you understand the basic concept of OOP Java, it would be easy to master.

- Secure :
  With Java's secure feature it enables to develop virus-free, temper-free systems. Authentication techniques are based on public-key

encryption.

- **Architecture-neutral :**

  Java compiler generates an architecture-neutral object file format, which makes the compiled code executable on many processors, with the presence of Java runtime system.

- **Portable :**

  Being architecture-neutral and having no implementation dependent aspects of the sepecification makes java portable.
  The compiler in Java is written in ANSI C with a clean portability boundary, which is a POSIX subset.

- **Robust :**

  Java makes an effort to eliminate error-prone situations by emphasizing mainly on compile time error checking and runtime checking.

- **Multithreaded :**

  With Java's multithreaded feature it is possible to write programs that can perform many tasks simultaneously.
  This design feature allows the developers to construct interactive

applications that can run smoothly.

- Interpreted :

  Java byte code is translated on the fly to native machine instructions and is not stored anywhere.
  The development process is more rapid and analytical since the linking is an incremental and light-weight process.

- High Performance :

  With the use of Just-In-Time compilers, Java enables high Performance.

- Distributed :

  Java is designed for the distributed environment of the internet.

- Dynamic :

  Java is considered to be more dynamic than C or C++ since it is designed to adapt to an evolving environment.
  Java programs can carry an extensive amount of run-time information that can be used to verify and resolve accesses to objects at run-time.

5) Define following.

(1) Byte Code

- Java Byte code is the machine code in the form of a .class file.
- with the help of this we can achieve platform independence in java.

(2) Java Virtual Machine (JVM)

- JVM is an abstract machine. It is a specification that provides runtime environment in which java bytecode can be executed.

(3) Java Runtime Environment (JRE)

- JRE is a set of software tools for development of Java applications. It combines the JVM, platform core classes and supporting libraries.

(4) Java Development Kit (JDK)

- JDK is a software development environment used for developing Java applications and applets. It includes the JRE, and interpreter/loader (Java), a compiler (Javac), an archiver (jar), a documentation generator (javadoc) and other tools ~~to~~ needed in java development.

6) Explain Static keyword.

- The static keyword in Java is used for memory management mainly.
- we can apply java static keyword with variables, methods, blocks and nested class.
- The static keyword belongs to the class than an instance of the class.

The static can be:

1.) Variable (also known as class variable)
2.) Method (also known as class method)
3.) Block
4.) Nested class

* Java Static Variable

If you declare any variable as static, it is known as a static variable.

- The static variable can be used to refer to the common property of all objects (which is not unique for each object). for example, the company name of employees, college name of students, etc.

- The static variable gets memory only once in the class area at the time of class loading.

* Static Method

- When a method is declared with static keyword, it is known as static method.
- The most common example of a static method is main() method.
- Methods declared as static have several restrictions:

• They can only directly call other static methods.
• They can only directly access static data.
• They cannot refer to this or super in any way.

* Static Blocks

- If you need to do computation in order to initialize your static variables, you can declare a static block that gets executed exactly once, when the class is first loaded.

* Static nested class

- we can not declare top-level class with a static modifier, but can declare nested classes as static.

Page No.:

Date:

- Such type of classes are called Nested static classes.

8/2/2020