

Department of Computer Science
Course Title: Programming in Java
Course Code: CC-3

Sri Ramkrishna Sarada Vidyamahapitha
Kamarpukur, Hooghly

1) What is Object Oriented Programming? How it is different from Procedural concepts?

Answer:

Object-oriented programming: As the name suggests, Object-Oriented Programming or OOPs refers to languages that uses objects in programming. Object-oriented programming aims to implement real-world entities like inheritance, hiding, polymorphism etc in programming. The main aim of OOP is to bind together the data and the functions that operate on them so that no other part of the code can access this data except that function.

SL. NO	PROCEDURAL ORIENTED PROGRAMMING	OBJECT ORIENTED PROGRAMMING
1.	In procedural programming, program is divided into small parts called functions .	In object oriented programming, program is divided into small parts called objects .
2.	There is no access specifier in procedural programming.	Object oriented programming have access specifiers like private, public, protected etc.
3.	Procedural programming follows top down approach .	Object oriented programming follows bottom up approach .
4.	Procedural programming does not have any proper way for hiding data so it is less secure .	Object oriented programming provides data hiding so it is more secure .
5.	Examples: C, FORTRAN, Pascal, Basic etc.	Examples: C++, Java, Python, C# etc.

2) "Java is called Machine Independent language" - Justify this statement with proper explanation.

Answer:

JVM translates bytecode into machine language

Every Java program is compiled into an intermediate language called Java bytecode. The JVM is used to both translate the bytecode into the machine language for a particular computer, and actually execute the corresponding machine-language instructions as well. The JVM and bytecode combined give Java its status as a "portable" language.

Machine language is OS dependent

Given the previous information, it should be easier to deduce an answer to the question. Since the JVM must translate the bytecode into machine language, and since the machine language depends on the operating system being used, it is clear that the JVM is platform (operating system) dependent. This fact can be verified by trying to download the JVM – you will be given

a list of JVM's corresponding to different operating systems, and you will obviously pick whichever JVM is targeted for the operating system that you are running.

3) Explain the important features of Java.

Answer:

Java is a platform independent language.

Platform independence means that we can write and compile the java code in one platform (e.g. Windows) and can execute the class in any other supported platform e.g. (Linux, Solaris, etc.).

1) Simple, Small and familiar

Java is a simple and small language. The Syntax of java is just like c++, so it is very easy to learn. It is simple because it

- i) does not use header files
- ii) eliminated the use of pointer
- iii) operator overloading and virtual base classes are eliminated.

2) Object oriented

Java is a pure Object oriented. Everything in java is object. All programs and data reside inside objects and classes

3) Distributed

Java has networking facilities. So java can create application on network.

4) Robust

Java gives importance to memory management by using the technique called Garbage Collection and Exception handling.

5) Secure

Since java is used on internet, security is an important issue. A security code is asked before a java code is interpreted on internet.

6) Platform independent

Java compiler generates a platform independent code called bytecode.

7) Portable

The Byte code generated by java can be used on any machine. So it can be portable.

8) Compiled and interpreted

Generally computer languages are either compiled or interpreted. but java combines both compiler and interpreted.

9) High performance

The use of byte code makes the performance high. The speed is also high with comparing c, c++.

10) Multithreading and interactive

Multithreading means handling more than one job at a time. Java supports Multithreading.

11) Dynamic and extensible

Java is a dynamic language. So it is capable of linking dynamic new classes, methods and

objects. Java supports functions written in c and c++ also. These functions are called native methods. During Run-Time Native methods can be linked dynamically.

4) What do you understand by Java virtual machine?

Answer:

Java Virtual Machine is a virtual machine that enables the computer to run the Java program. JVM acts like a run-time engine which calls the main method present in the Java code. JVM is the specification which must be implemented in the computer system. The Java code is compiled by JVM to be a Bytecode which is machine independent and close to the native code.

5) What is the difference between JDK, JRE, and JVM?

Answer:

JVM

JVM is an acronym for Java Virtual Machine; it is an abstract machine which provides the runtime environment in which Java bytecode can be executed. It is a specification which specifies the working of Java Virtual Machine. Its implementation has been provided by Oracle and other companies. Its implementation is known as JRE.

JVMs are available for many hardware and software platforms (so JVM is platform dependent). It is a runtime instance which is created when we run the Java class. There are three notions of the JVM: specification, implementation, and instance.

JRE

JRE stands for Java Runtime Environment. It is the implementation of JVM. The Java Runtime Environment is a set of software tools which are used for developing Java applications. It is used to provide the runtime environment. It is the implementation of JVM. It physically exists. It contains a set of libraries + other files that JVM uses at runtime.

JDK

JDK is an acronym for Java Development Kit. It is a software development environment which is used to develop Java applications and applets. It physically exists. It contains JRE + development tools. JDK is an implementation of any one of the below given Java Platforms released by Oracle Corporation:

- Standard Edition Java Platform
- Enterprise Edition Java Platform
- Micro Edition Java Platform

6) What is the output of the following Java program?

1. **class** Test
2. {
3. **public static void** main (String args[])
4. {

```
5.     System.out.println(10 + 20 + "Javatpoint");
6.     System.out.println("Javatpoint" + 10 + 20);
7.     }
8.     }
```

Answer:

The output of the above code will be

```
30Javatpoint
Javatpoint1020
```

Explanation

In the first case, 10 and 20 are treated as numbers and added to be 30. Now, their sum 30 is treated as the string and concatenated with the string **Javatpoint**. Therefore, the output will be **30Javatpoint**.

In the second case, the string Javatpoint is concatenated with 10 to be the string **Javatpoint10** which will then be concatenated with 20 to be **Javatpoint1020**.

7) Explain the various use of static and final keyword in java.

Answer:

Java - final keyword:

Final keyword defines itself that once final keyword is used then one cannot extend or change its value. In java the final keyword is used in different methods to define any variable that can be only assigned one time in program.

Use of final keyword:

The final keyword has mainly three uses one of them is to create final class. Second one is to use final methods and third one is to use final data member.

Following are uses final keyword:

1. Stop Inheritance.
2. Stop Method overriding.
3. Stop value change.

final class:

It is used to avoid inheritance. Once a final word assigned before class then this class cannot be inherited further. In other words final class cannot have its derived class.

8) Write Short note on the following: Static Variable vs. Static Method

Answer:

Static Variable in Java:

Static variable in Java is variable which belongs to the class and initialized only once at the start of the execution.

- It is a variable which belongs to the class and not to object(instance)
- Static variables are initialized only once, at the start of the execution. These variables will be initialized first, before the initialization of any instance variables
- A single copy to be shared by all instances of the class
- A static variable can be accessed directly by the class name and doesn't need any object

Static Method in Java:

Static method in Java is a method which belongs to the class and not to the object. A static method can access only static data.

- It is a method which belongs to the class and not to the object(instance)
- A static method can access only static data. It can not access non-static data (instance variables)
- A static method can call only other static methods and can not call a non-static method from it.
- A static method can be accessed directly by the class name and doesn't need any object
- A static method cannot refer to "this" or "super" keywords in anyway