**JAVA INTERVIEW QT AND ANS**

**1.Tell us something about JIT compiler.**

ANS = .java file is compiled into .class file/byte code and not executable directly by any underline m/c.....

Java gives us an java virtual machine (JVM) which converts byte code to operational code(OP Code ) for underline machine

**2.What do you understand by an instance variable and a local variable?**

ANS = The variables we declare whose requirement is only in that particular method are called as local variables. Whereas is we require some variables in more than 1 method, those variables are called as instance variables. Instance variables are declare in different class than method. By more than 1 method we can point to these variables. Whereas local variables are declared in method

**3. Why is Java not a pure object oriented language?**

ANS = Java is not fully object oriented because it supports primitive data type like Int , byte, long etc. ,which are not objects. Because in JAVA we use data types like int, float, double etc which are not object oriented, and of course is what opposite of OOP is. That is why JAVA is not 100% objected oriented.

**4.Pointers are used in C/ C++. Why does Java not make use of pointers?**

ANS = Pointers are quite complicated and unsafe to use by beginner programmers. Java focuses on code simplicity, and the usage of pointers can make it challenging. Pointer utilization can also cause potential errors. Moreover, security is also compromised if pointers are used because the users can directly access memory with the help of pointers. Thus, a certain level of abstraction is furnished by not including pointers in Java. Moreover, the usage of pointers can make the procedure of garbage collection quite slow and erroneous. Java makes use of references as these cannot be manipulated, unlike pointers.

**5. Why is Java a platform-independent language?**

ANS = JIT stands for Just-In-Time and it is used for improving the performance during run time. It does the task of compiling parts of byte code having similar functionality at the same time thereby reducing the amount of compilation time for the code to run.

**6. Explain access modifiers in Java.**

ANS = There are two types of modifiers in Java: access modifiers and non-access modifiers.

The access modifiers in Java specifies the accessibility or scope of a field, method, constructor, or class. We can change the access level of fields, constructors, methods, and class by applying the access modifier on it.

There are four types of Java access modifiers:

Private: The access level of a private modifier is only within the class. It cannot be accessed from outside the class.

Default: The access level of a default modifier is only within the package. It cannot be accessed from outside the package. If you do not specify any access level, it will be the default.

Protected: 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.

Public: The access level of a public modifier is everywhere. It can be accessed from within the class, outside the class, within the package and outside the package.

**7. What is typecasting?**

ANS = Type casting is when you assign a value of one primitive data type to another type. In Java, there are two types of casting: Widening Casting (automatically) - converting a smaller type to a larger type size. byte -> short -> char -> int -> long -> float -> double.

Narrowing Casting (manually) - converting a larger type to a smaller size type.

double -> float -> long -> int -> char -> short -> byte

**8. Which Java IDE to use, and why?**

ANS = Eclipse is a Java IDE that is one of the 3 biggest and most popular IDE’s in the world. It was written mostly in Java but it can also be used to develop applications in other programming languages apart from Java using plug-ins.

PDE (Plugin Development Environment) is available in Eclipse for Java programmers that want to create specific functionalities in their applications.

Eclipse flaunts powerful tools for the various processes in application development such as charting, modeling, reporting, testing, etc. so that Java developers can develop the application as fast as possible.

Eclipse can also be used to create various mathematical documents with LaTeX using the TeXlipse plug-in as well as packages for the Mathematica software.

Eclipse can be used on platforms like Linux, macOS, Solaris and Windows.

Advantages of IDE:

1. Using IDE will cost you less time and effort .

2.Navigation is made easier.

3.Auto completion- one of the best features , you don’t have to remember all.

4.Refactoring

5.Error debugging is easy , you can easily navigate to Error line.

6.All files can be viewed and managed at same screen.

7.Organizing you imports.

8.Downloading requires packages at ease.

In addition to these specific Eclipse advantages are:

• It is free and open source.

• Industrial level of development

• It supports many other languages other than JAVA.

• Framework integration like Junit and TestNG and other plugins can be done easily.

• Since Eclipse tools are open source, they are quickly updated with the latest technology that can be integrated into existing code.

**9. Differentiate between JVM, JRE, and JDK**

ANS = JDK(Java development kit) is a complete package of tools which needed to compile and run Java program. It consists of Java compiler , JRE and also documentation .

JRE stands for Java Run time Environment which provide an environment to execute Java program.if we just want to execute a program or Java .class file , we just need to install JRE and not JDK because no compilation is required .

JDK = JRE + Javac + libraries

JRE = JVM + other libraries

**10. Explain about Java Virtual Machine?**

ANS = JVM (Java Virtual Machine) is an abstract machine. It is a specification that provides runtime environment in which java bytecode can be executed.JVMs are available for many hardware and software platforms (i.e. JVM is platform dependent).

**A specification** where working of Java Virtual Machine is specified. But implementation provider is independent to choose the algorithm. Its implementation has been provided by Oracle and other companies.

**An implementation** Its implementation is known as JRE (Java Runtime Environment).

**Runtime Instance** Whenever you write java command on the command prompt to run the java class, an instance of JVM is created.

**11. Encapsulation :**

Encapsulation is the process of combining data and functions into a single unit called class. In Encapsulation, the data is not accessed directly; it is accessed through the functions present inside the class. In simpler words, attributes of the class are kept private and public getter and setter methods are provided to manipulate these attributes. Thus, encapsulation makes the concept of data hiding possible.(Data hiding: a language feature to restrict access to members of an object, reducing the negative effect due to dependencies. e.g. "protected", "private" feature in Java).

**12 .Inheritance :**

Inheritance is a process in which one object acquires all the properties and behaviors of its parent object automatically. In such a way, you can reuse, extend or modify the attributes and behaviors which are defined in other classes.

In Java, the class which inherits the members of another class is called derived class and the class whose members are inherited is called base class. The derived class is the specialized class for the base class.

Types of Inheritance :

1. Single inheritance : When one class inherits another class, it is known as single level inheritance

2. Hierarchical inheritance : Hierarchical inheritance is defined as the process of deriving more than one class from a base class.

3. Multilevel inheritance : Multilevel inheritance is a process of deriving a class from another derived class.

**13. Types of Polymorphism IMP**

1. Compile Time Polymorphism (Static)

2. Runtime Polymorphism (Dynamic)

Let’s understand them one by one :

1.Compile Time Polymorphism : The polymorphism which is implemented at the compile time is known as compile-time polymorphism. Example - Method Overloading

Method Overloading : Method overloading is a technique which allows you to have more than one function with the same function name but with different functionality. Method overloading can be possible on the following basis:

1. The return type of the overloaded function.

2. The type of the parameters passed to the function.

3. The number of parameters passed to the function.

2. Runtime Polymorphism : Runtime polymorphism is also known as dynamic polymorphism. Function overriding is an example of runtime polymorphism. Function overriding means when the child class contains the method which is already present in the parent class. Hence, the child class overrides the method of the parent class. In case of function overriding, parent and child classes both contain the same function with a different definition. The call to the function is determined at runtime is known as runtime polymorphism.

**14. super**

1. super is used to give a call to an overridden method of the superclass

2. super is used to give a call to the superclass constructor

3. superclass parameterless constructor is called automatically via subclass parameterless constructor

so the writing of super() is optional in the derived class constructor

4. superclass parameterized constructor is called with the help of super(list of parameters) inside sub class parameterized constructor

5. super() statement must be the first statement in the method