ASSIGNMENT NO. 1

Java Interview Questions

Name : Pushpak R. Gulkari

Q. 1. What do you know about JVM, JRE & JDK ?

→ JVM stand for Java Virtual Machine. JVM is responsible for interpreting the bytecode and translating it into machine code that can be executed by the underlying O.S. It also provides memory management, security and other runtime services necessary for executing Java applications.

JRE :-

JRE stands for Java Run Time Environment. JRE is a platform dependent software that we can download from oracle website. We must install JRE on client's Machine to run Java application.

JRE = rt.jar + Java Virtual Machine.

We can execute java file by JRE on any system.

JDK :- JDK stands for Java Development Kit. Developer must install JDK to develop Java application. JDK is a platform dependent software. JDK comprises of Java Development Tools, Java DOIs, rt.jar, Java Virtual machine.

Q. 2. Is JRE platform dependent or independent?

→ JRE provides the system with the min. requirements or environment needed for the execution of a Java program in system.

JRE is platform dependent as the configuration of each OS is different from each other.

for example → Mac OS → own ~~JRE~~ JVM
Window → own ~~JRE~~ JVM
Linux → own ~~JRE~~ JVM.

Q. 3. Which is ultimate base class in java class hierarchy? List the name of methods of it?

→ Class object is the root of ~~the~~ class ~~hierarchy~~ hierarchy. Every class has object as a superclass.

Methods :-

String toString() → Returns a string representation of the object

Int hashcode() → Returns a hash code value for object.

Boolean equals (Object obj) → Indicates whether some other object is equal to this one.

Q.4. Which are the reference types in java?

→ Variable of non-primitive data type contains reference of instance hence non-primitive data types also called as reference types. Types →

1. Interface.
2. class
3. Enum
4. Array.

Q.5. Explain narrowing and widening?

→ Narrowing :-

Assignment Conversion of larger data type variable in smaller data type variable is called as Narrowing. In narrowing explicit typecasting is necessary.

e.g. int num = (int) div
↓
float or long or double.

Widening :-

Assignment Conversion of smaller data type to larger data type is called as widening. Typecasting is optional.

e.g. double abc = y
↳ int.

Q.6. How will you print "Hello CDAC" statement on Screen, without semicolon?

→

```
if((System.out.print("Hello CDAC")
        == 0)
```

Q.7. Can you write Java application without main function? If yes, how?

→ Yes we can write & execute a java program without a main method by using a static block. Static block gets executed only once when a class is loaded and does not depend on main method.

Q.8. What will happen, if we call main method in static block?

→ If we call main method in static Block the main method will not run.

Q.9. In System.out.println, Explain meaning of every word.

→ "out" is a reference variable of ~~public final static~~ PrintStream class which is declared as public final static and has value 'null'. And out is declared in "System" class which is public final in java.lang package. "println" is a non static method of java.io.PrintStream

class & gives output as string followed by new line.

Q. 10. How will you pass the object to the function by reference?

→ It can be done by passing the object of the class in the place of the actual parameter and the formal parameter of a class object type has the same reference to each other that's why with the help of the formal parameter object of class any changes will be reflected in both object formal and actual ~~reference~~ address.

Q. 11. Explain constructor chaining. How can we achieve it in C++?

→ Constructor chaining refers to the ~~the~~ concept which can be described as calling of another constructor from a current constructor.

```
Employee ( ) {
    empid ( 17);
}
Employee ( int x) {
    empid = x
}
```

Q.12. Which are the rules to overload method in subclass?

→ (1) We have to give different no. of parameters
(2) If no. of parameters is same then data type of parameters must be differ
(3) If no. of parameters & data types are also same, then order of parameters must be different.
(4) Access specifier in subclass should be less restrictive than base class.

Q 13. Explain difference among finalize and dispose?

→ finalize gets called by GC when this object is no longer in use. Dispose is just a normal method which the user of this class can call to release any resources. If user forgot to call Dispose and if the class have finalize implemented then GC will make sure it gets called.

Q.14. Explain the difference between final, finally and finalize.

→ final :- final keyword when used with variable, the value of variable cannot be reassigned, when used with class it cannot be inherited. When used with method it cannot be overridden.

Finally: Finally keyword is used in association with try/catch block and guarantees that a section of code will be executed, even if an exception is thrown.

finalize: finalize method in java is a method of object class that is used to perform cleanup activity before destroying any object. It is called by default by garbage collector before deletion of an object.

Q.15. Explain the difference among checked and Unchecked exception.

→ The difference is that a checked exception is caught at compile time whereas a runtime or unchecked exception is at runtime.

Q.16. Explain Exception Chaining:

→ A chained exception is an exception that is caused by another exception. chained exceptions are associated such that the previous exception causes each exception in the chain.

Q.17. Explain the difference among throw and throws.

→ The Throw keyword is used inside a method. whereas the Throws keyword is used in the method signature. The Throw Keyword throws an exception explicitly, whereas the Throws keyword declares that a method might throw exception.

Q.18. In which case finally block doesn't execute?

→ The finally block may not execute if the JVM exits while the try or catch code is being executed. The try block of the writeList method that you've been working with here opens a PrintWriter. The program should close that stream before exiting the writeList method.

Q.19. Explain ~~Do~~ Upcasting.

→ Process of converting reference of sub class into reference of super class is called as upcasting. Super class reference can contain reference of sub class instance. If we want to minimize instance dependency in code then we should use upcasting.

Q.20. Explain dynamic method dispatch.

→ It allows Java to support overriding of methods which is central for runtime polymorphism. It allows a class to specify methods that will be common to all of its derivatives, while allowing subclasses to define specific implementation of some or all of those methods.

Q.21. What do you know about final method?

→ If we declare a method as final then it cannot be overridden but it can be overloaded. Final method cannot be modified.

Q.22. Explain fragile Base class problem and how can we overcome it?

→ The fragile Base class problem is fundamental architectural problem of OOPS where base classes are considered fragile because seemingly safe modifications to a base class, when inherited by the derived classes may cause the derived classes to malfunction.

To overcome this problem we should declare super class as final.

Q.23. Why java does not support multiple implementation inheritance?

→ The reason behind this is to prevent ambiguity. Consider a case where class B extends class A & class C and both A & C have the same method display(). Now java compiler cannot decide, which display method it should inherit. To prevent such situation, multiple inheritance is not allowed in Java.

Example → Diamond problem.

Q.24. Explain marker interface? List the name of marker interfaces.

→ An interface that does not have any methods fields or constants i.e. an empty interface in java is known as marker or tag interface. It is used to deliver type information at runtime to the JVM so that it can take some action based on the information received.

Example:

Serializable, Cloneable, Remote interface etc.

Q.25 Explain the significance of marker interface.

→ It provides runtime information about objects, so the compiler and JVM have additional information about the object.

Marking interface makes an object as remote, which is accessible by another machine (host machine).