Core Java selective Question & Answer

Q-1: What do you mean by Class and Object?
Answer:

A class denotes a category of objects and acts like a blue print for creating objects. That means a class is a collection of objects that have common properties and behavior and relationship.

Instance of a class is called an object. An object is a collection of variable or properties and methods.

Object = variable/properties + methods.

Q-2: What do you mean by encapsulation?
Answer:

Encapsulation is a process that allows selective hiding of properties and methods within objects. This is achieved by specify an object as private in the definition of the class. The main advantage of encapsulation is the ability to hide the implementation of a class.

Q-3: What do you mean by inheritance?
Answer:

Inheritance is the property that allows the reuse of an existing class to build a new class. The inherited class is called base class or super class and the class that inherits the base class is called sub class/derived class.

Q-4: What are the benefits of Inheritance?
Answer:

One of the key benefits of inheritance is to minimize the amount of duplicate code in an application by sharing common code amongst several subclasses.
Inheritance can also make application code more flexible to change because classes that inherit from a common super class can be used interchangeably.

Q-5: What do you mean by narrowing and widening? Give example in each case.
Answer:

Narrowing: Converting a broader data type into a narrower data type is called narrowing, which results in loss of magnitude information.

```
Ex:   double d = 10.55555;
      int x = (int)d;
```

Widening: Converting a narrower data type to a broader data type without loss of information is called widening.

```
Ex:   int x =3, y = 2;
      Double d =(double)x/y;
```

Q-6: What do you mean by "super ()" and "this"?
Answer:

"super ()" refers to invoking of super class constructor. And "this" refers to current class object.

Q-7: What do you mean by method overloading and method overriding?
Answer:

Overloading: A class having two or more methods with same name but with different arguments then we say that those methods are overloaded. Compile time or static polymorphism can be achieved using method overloading.
Example:

```
int sum(int x, int y) {
        //bodypart;
}
Int sum(int x, int y, int z){

        }
```

Overriding: Define two or more methods with the same name and same number, type and sequence of parameter in a derived class is called method overriding. Runtime or dynamic polymorphism can be achieved using method overriding.

Example:

```
interface A{
int sum(int x, int y);
}
```

```
class B implements A{
public int sum(int x, int y){
        }
}
```

Q-8: What is the difference between Interface and Abstract class?
Answer:

Interface	***Abstract class***
Java interface should be implemented using keyword “implements”;	A Java abstract class should be extended using keyword “extends”.
A Java class can implement multiple interfaces	It can extend only one abstract class.
Variables declared in a Java interface is by default final.	An abstract class may contain non-final variables.
Methods does not contain body part	Methods may or may not contain body part

Q-9: What do you mean by polymorphism?
Answer:
Polymorphism enables the same method to behave differently on different class.

Q-10: What is JVM?
Answer:
JVM stands for Java Virtual Machine. This is nothing but a software. JVM creates a run time system internally that helps the execution of code by –

(a) Loading the class file.
(b) Loading the byte code.
(c) Performing the garbage collection.

***Q-11: Define package. What are the advantages of package? Write down the name of* default name of package.**
Answer:
A package in java is an encapsulation mechanism that can be used to group related class, interfaces and sub packages.
Advantages of package:

a. Package allows us to organize our class into smaller units and make it easy to locate and use the appropriate class file.
b. It helps us to avoid naming conflict.
c. It protects our class and methods in a larger way than a class to class basis.
d. Package name can be used to identify our classes.

The default package name in java is java.lang

Q-12: What is the use of abstract, final and static keyword?
Answer:
Use of abstract:
a. With class - indicates that the class cannot be instantiate.
b. With method - indicates that the method must be overridden in the sub class of that abstract class.

Use of final:
a. With class - indicates that the class cannot be subclass
b. With variable - indicates that the assigned value cannot be change.
c. With method - indicates that method body cannot be changed.
d. With object of a class - indicates that object reference cannot be changed but value can be changed.

Use of static keyword:
a. static variables of a class are accessed by static methods only.
b. static methods of a class are invoked by static methods only.
c. static method does not have “this”.

Q-13: What do you mean by logical and short-circuit operator
Or, what is the difference between & and &&?
Answer:

The difference between & and && is that the conditional or short-circuit operator (&&) will not bother to evaluate the right hand operand if the left hand operand is false whereas, the logical operator (&) will evaluate both the operands.

Q-14: What is a constructor? What do you mean by default constructor?

Answer:

Constructor:

If a method in a class have the same name as the class name and must not return a value than we can consider the special method as a constructor. The primary purpose of a constructor is initializing the instant variable.

Default constructor:

If we do not define any constructor for a class the compiler will supply a default constructor in the class, which actually does nothing. The default constructor is also described as no argument constructor. Because it requires no argument to be specified when called. When we create an object of a class the default constructor is automatic invoked.

Ex: TestClass tc = new TestClass();

Q-15: What is an Array? How many ways can create an Array in java?

Answer:

Array:

An Array in Java is an object that is a named set of variable of the same type. Each variable is called array element.

We can declare an array in several ways.

```
int[] arr = new int[5];
int arr[] = new int[5];
int[] arr = {0,1,2,3,4};                    //dense array
int[][] arr2 = new int[length1][length2];          //2D array
Object list1 = new Object[5];       //An array of objects
```

Q-16: Write down ten keywords in Java.

Answer:

Short, abstract, static, break, case, catch, final, extends, private, import, switch, synchronized, native, transient, volatile, while, instanceof, goto, enum, throws etc.

Q-17: What is the difference between primitive data type and wrapper class?

Answer:

Java provides eight primitive data types. Primitive data types in java are not object. In order to manipulate this values/data types as objects the java.lang package provides a wrapper class for each of this primitive data types. All wrapper classes are final.

Primitive Data Type	Wrapper Class
boolean	Boolean
byte	Byte
char	Character
short	Short
int	Integer
long	Long
float	Float
double	Double

Q-18: What is the difference between local variable, instance variable and static variable?

Answer:

Local variable:

All the variables declared inside a method are called local variable. Local variable is also called stack variable because they live on stack.

Instance variable:

The variables that declared inside a class but outside any methods which have global scope are called instance variable. Instance variable belongs to object and this kinds of a variable lives on heap. Instance variable are also called non-static variable.

Static variable:

The variable which are declared with a modifier “static” and which scope belongs to class are called static variable.

Q-19: What is the use of finalize () and finally block?

Answer:

The automatic garbage collector calls the finalized () method that is eligible for garbage collector before actual destroying the object.

A finally block is an optional block that exist after the last catch block and always executed whether or not exception is caught.

Q-20: Java supports how many types of access modifier explain them.

Answer:

Java supports four types of access modifier.

a. public
b. protected
c. private
d. default

Accessibility Criteria:

Modifier	Same Class	Same Package	Subclass	Universe
private	yes			
default	yes	yes		
protected	yes	yes	yes	
public	yes	yes	yes	yes

Q-21: What is the difference between ‘equals ()’ methods and ‘==’ operator.?

Answer:

equals ():

equals () method checks the equality of the content.

== :

“==” checks the equality of object reference.

Q-22: Explain the difference between pass by value and pass by reference.

***Answer*:**

In pass by value, a copy of variable is passed to the method; the values of the variable can’t be changed in the method.

In pass by reference, a pointer of the variable is passed and the values can be modified in the method.

Q-23: What is an Exception?

Answer:

An event that occurs during the execution of a program that disrupts the normal flow of instructions is called an exception.

Q-24: Explain the purpose of garbage collection that the JVM uses.

Answer:

Garbage collection in Java identifies and discards the objects that are no longer needed by a program so that their resources can be reclaimed and reused. A Java object is subject to garbage collection when it becomes unreachable to the program in which it is used. The finalize () method is called by garbage collector when it determines no more references to the object exists.

Q-25: Checked Exceptions vs. Unchecked Exceptions.

Answer:

Checked Exceptions

- A checked exception is subclass of Exception excluding class RuntimeException and its subclasses.
- Compiler checks to see if these exceptions have been properly caught or not. Else the code doesn’t compile. Thus, a program is forced to deal with the situations where an exception can be thrown.
- Checked exceptions must be either declared or caught at compile time.

Unchecked Exceptions

- Unchecked exceptions are RuntimeException and all of its subclasses along with class java.lang.Error and its subclasses also are unchecked.
- A program does compile without these exceptions being handled during compile time.

Q-26: What do you mean by Assertion?

Answer:

Assertions are simple check assumption made at the beginning of the program to ensure the program is true throughout provided by the Java language. For example, the range of age should be in between 18 and above; or cannot be more than 50.

An Assertion contains a Boolean expression which is set to be true in the beginning of the program.
Syntax:
assert<Boolean exp.>; // simple form
assert <Boolean exp.>:<message exp.>; //augmented form

Q-27: What do you mean by* Inner class*?

Answer:

An inner class is associated with an instance of its enclosing class and has direct access to that object's methods and fields. Also, because an inner class is associated with an instance, it cannot define any static members itself.
Objects that are instances of an inner class exist *within* an instance of the outer class. Consider the following classes:

```
class OuterClass {
   ...
   class InnerClass {
      ...
   }
}
```

Q-28: What do you mean by auto boxing?

Answer:

The automatic conversion of primitive int type into a wrapper class object is called auto boxing. It does not require to type cast the int value. The modification of primitive wrapper objects is done directly. The following example illustrates auto boxing:

```
int number;
Integer intObject;
number = 1;
intObject = 2;
number = intObject;
intObject = number;
```

Q-29: What is type casting?

Answer:

Type casting means to explicitly convert one type to another.
For example, the following lines:

```
double x = 5.0;
int y = x;
```

will produce an error message, because of a possible loss of precision (any decimals will get lost when converting to int). The following will work however:

```
double x = 5.0;
int y = (int) x;
```

The programmer is forcing the Java compiler to accept the conversion; saying, in effect: "Please do this anyway, I know what I am doing".

Q-30: What do you mean by instanceof operator?

Answer:

The instanceof operator is used to check the type of an instance of an object.

```
Ex:
  String s = "XYZ";
  if (s instanceof java.lang.String)
        returns TRUE.
```