**Core Java**

Basics & Oops

1. What is Java?

Java is an object oriented**programming language which follows the concepts of OOPS like Abstraction, polymorphism, inheritance and encapsulation.**

1. What is the Java memory Model?

* Class Area: It consists of the static variables.
* Heap: The objects created at the run time are stored in heap.
* Stack: It holds the primitive variables, partial results and local variables.

Program Counter: It consists of the address of the current instruction being executed.

1. What are the types of variables in JAVA?

* Local Variables: These variables are declared within a method and their scope is also within the method. They don’t have any default values.
* Static Variables: The variable declared as static are known as static variables. It cannot be local.
* Instance Variables: The variables declared outside a method but inside a class are known as instance variables. They do have default values like int (0), string (null) etc.

What is class and object?

Class is a blueprint or a template from which objects are collected. That is why objects are known as instances (result) of class.

1. What is method overloading and overriding?

Method overloading is the **concept** of a class having multiple methods of same name but vary with type and number of arguments. It enhances the readability. It is performed within a class.

Method overriding is the **concept** of implementing a method in the sub class which is already provided in the super class. It should compulsorily be in an IS-A relation. The subclass method should have the same method name and the parameters

1. What is static and dynamic binding?

The **concept** of Establishing a connection between method body and method call is known as binding. There are two types of binding.

* Static Binding
* Dynamic binding

Static binding: The **concept** of determining the object at the compile time then it is known as early binding or static binding. When a class has final, static or private method is also known as static binding.

Dynamic binding: The **concept** of determining the type of the object at the run time it is known as dynamic binding.

6)What are different types of access modifiers?

Public: Accessible everywhere.

Private: Accessible within the class.

Protected: Accessible within the package and outside the package only through inheritance.

Default: Accessible only within the package.

7)What do you mean by platform independence of Java?

Platform independence means that you can run the same Java Program in any Operating System. For example, you can write java program in Windows and run it in Mac OS. JVM is responsible for converting byte code into machine readable code where any OS can understand but JVM is not platform independent, thats why you have different JVM for different operating systems.

### 8)Can we overload main method?

Yes, we can have multiple methods with name “main” in a single class. However if we run the class, java runtime environment will look for main method with syntax as public static void main(**String** args[]).

### Can we have multiple public classes in a java source file?

We can’t have more than one public class in a single java source file. A single source file can have multiple classes that are not public.

### What is Java Package and which package is imported by default?

Java package is the mechanism to organize the java classes by grouping them. The grouping logic can be based on functionality or modules based. A java class fully classified name contains package and class name. For example, java.lang.Object is the fully classified name of Object class that is part of java.lang package.java.lang package is imported by default and we don’t need to import any class from this package explicitly.

### What is Marker interface?

A marker interface is an empty interface without any method but used to force some functionality in implementing classes by Java. Some of the well known marker interfaces are Serializable and Cloneable.

### What are Wrapper classes?

Java wrapper classes are the Object representation of eight primitive types in java. All the wrapper classes in java are immutable and final. Java 5 autoboxing and unboxing allows easy conversion between primitive types and their corresponding wrapper classes.

Strings

What is string pool?

String pool is a pool of strings stored in java heap memory. String is possible only because of the immutability concept of the string. String pools help in saving lot of space for java Runtime. When we use double quotes to create a String, it first looks for String with same value in the String pool, if found it just returns the reference else it creates a new String in the pool and then returns the reference.

4) Why string is immutable?

Majority of the data is represented in the form of strings. As a result, there are high chances of strings pointing to the same reference. If a string changes, it will affect the other string which is being referenced to the same pointer and there are chances that it might become unreferenced which gets deleted by the automatic garbage collector. This is the reason why strings are made immutable.

25) How do you make a class immutable?

By declaring a class final.

By declaring the properties of the class as final.

By not providing the setters.

26)StringBuffer vs StringBuilder ?

|  |  |  |
| --- | --- | --- |
| **No.** | **StringBuffer** | **StringBuilder** |
| 1) | StringBuffer is *synchronized* i.e. thread safe. It means two threads can't call the methods of StringBuffer simultaneously. | StringBuilder is *non-synchronized* i.e. not thread safe. It means two threads can call the methods of StringBuilder simultaneously. |
| 2) | StringBuffer is *less efficient* than StringBuilder. | StringBuilder is *more efficient* than StringBuffer |

27) == vs Equals?

== method checks for the reference of the string.

Equals method checks for the actual content of the string.

Exception Handling

What is exception handling and how do you achieve it?

Exception handling is a problem that occurs during the execution of the program. It disrupts the normal flow of the program and the application terminates abnormally, which is not recommended so these exceptions are to be handled. It is achieved through try, catch and throw blocks.

)How do you make sure a code must be executed even if exception happens?

By placing the code int the finally block.

30)What code you normally write in finally block?

Most important statements like closing connections and statements are written in finally block. It should always be followed by try or catch block. The finally block gets executed even the exception is handled or not.

31)What are checked vs unchecked exceptions?

Checked exceptions are exceptions that occur at the compile time. The extend only the throwable class. Example: File not found exception.

Unchecked exceptions are exceptions that occur at the run time. For example: - array out of bounds exception.

32)How do you create custom exceptions?

Using the throw keyword, we can explicitly throw exceptions either checked or unchecked exceptions. It is mainly used to throw custom exceptions.

33)How does exception propagation works?

An exception is first thrown from the top of the stack and if it is not caught, it drops down the call stack to the previous method, if not caught there, the exception again drops down to the previous method, and so on until they are caught or until they reach the very bottom of the call stack. This is called exception propagation.

34)exception vs error?

|  |  |
| --- | --- |
| Errors | Exceptions |
| Errors in java are of type java.lang.Error. | Exceptions in java are of type java.lang.Exception. |
| All errors in java are unchecked type. | Exceptions include both checked as well as unchecked type. |
| Errors happen at run time. They will not be known to compiler. | Checked exceptions are known to compiler where as unchecked exceptions are not known to compiler because they occur at run time. |
| It is impossible to recover from errors. | You can recover from exceptions by handling them through try-catch blocks. |
| Errors are mostly caused by the environment in which application is running. | Exceptions are mainly caused by the application itself. |
| Examples : java.lang.StackOverflowError, java.lang.OutOfMemoryError | Examples : Checked Exceptions : SQLException, IOException Unchecked Exceptions : ArrayIndexOutOfBoundException, ClassCastException, NullPointerException |

### What is try-with-resources in java?

One of the Java 7 features is try-with-resources statement for automatic resource management. Before Java 7, there was no auto resource management and we should explicitly close the resource. Usually, it was done in the finally block of a try-catch statement. This approach used to cause memory leaks when we forgot to close the resource.From Java 7, we can create resources inside try block and use it. Java takes care of closing it as soon as try-catch block gets finished.

### What is multi-catch block in java?

Java 7 one of the improvement was multi-catch block where we can catch multiple exceptions in a single catch block. This makes are code shorter and cleaner when every catch block has similar code.If a catch block handles multiple exception, you can separate them using a pipe (|) and in this case exception parameter (ex) is final, so you can’t change it.

### Can we have try without catch block?

Yes, we can have try-finally statement and hence avoiding catch block.

Collections