**Topics**

1). Solid principles

2). What is Single responsibility principle?

3). What is open closed principle?

4). What is Liskov Substitution Principle?

5). What is Interface segregation principle?

6). What you mean by immutable class?

7). What is dependency inversion method?

8). How to make class are immutable?

9). What is instance variable?

10). Customer interface in java?

11). What is volatile keyword?

12). Forname ()

13). What is java serialization?

14). What is synchronization ?

15). Types of mechanisms for synchronization?

16). What is thread in java?

17). Ways to achieve threads in java?

18). What is threading

19). Threading vs multi-threading

20). What is java memory model?

21). What is JVM?

22). Why java is simple?

23). Why Java is robust?

24). Why java is secure?

25). What is JDK?

26). What is JRE?

27). What is method overloading?

28). What is method overriding?

29). What are the features of OOPS?

30). Types of inheritance?

31). Main memory areas in java ?

32). What is completable future ?

33). What is the difference between == and equals() ?

34). What Class loader in java?

35). What is deep copy?

**Solution**

1). Solid principle?

The solid principle are the set of object oriented design principles that helps to create code as more maintainable flexible and reusable. Solid principles in java are :-

> Single Responsibility principle (SRP)

> Open-Closed principle (OCP)

> Liskov Substitution principle (LSP)

> Interface segregation Principle (ISP)

> Dependency Inversion Principle (DIP)

2). What is Single responsibility principle?

It states that each class should only have one responsibility.it makes the code more modular. And easy to understand and more maintainable.

Eg: when are making an application we are creating different classes for each tasks .for example we are providing configuration class only for write the configuration logic. And controller class only for write controls logic.

3).What is open closed principle?

It states that the software entities (such as class,modules,functions, etc.) should be open for extensions and closed for modification.

This means that once a class is written and working, it should not be changed to add new behavior. Instead that new functionality should be added through extention, by creating new subclasses or compositions without altering the existing code.

(composition means building complex objects or larger structure by complaining similar objects or components.)

4). What is Liskov Substitution Principle?

It states that the object of a superclass should be replaceable with object of its subclass without affecting the correctness of the program.

In simple terms if we have two classes Animal and lion . The lion inherits Animal class then the subclass lion have the properties and methods of its own and also having the features of the class Animal.so we should be able to use lion class without any causes.

5). What is Interface Segregation principle?

It defines that an interface consist of many methods ,but a class only needs a few of those methods. According to ISP we should break down that big interface into smaller so the class can choose only the functions they actually need. By Implementing this we can avoid the unnecessary dependencies and making code more modular and maintainable

6). What you mean by immutable class?

An immutable class in java is a class ,that’s instance cannot be modified after creation. Once an object of the immutable class is created ,its state remains constant through out its life cycle. The immutable classes are thread safe .

7). What is dependency inversion method?

It states that two condition :-

->. High level modules should not depends on low level modules .Both should depends on abstraction.

->. Abstraction should not depends on details .Details should depend on abstraction.

8). How to make class are immutable?

> make all fields are private and final

> provide a constructor for initialize all the fields

> do not provide any methods that modifies the state of the object

9). What is instance variable?

Instance variable in java is a variable that declare inside the class but outside of the method. Each instance has its on copy of that variable.

10). Customer interface in java?

The customer interface in java is a functional interface that accept a single input argument and returns no results . It is the part of the java.utils.function package which was introduced in java 8 for supporting functional programming. The customer interface has one Abstract method :- accept ().it takes single argument and performs an operation on it .

11). What is volatile keyword?

The volatile keyword is mostly used in multi-threaded application to ensure that shared data is always consistent. It is a modifier that can be applied to variables to indicate that their values can be changed by different threads. It means that when a thread reads a volatile variable, It will always see the most up-to-date values, even if the values are changed by another thread.

12). Forname()

The forname method of java Class returns the class object associated with the class or interface with the given name in the parameter as string.

13). What is java serialization?

Serialization is the process of converting an object state to a byte stream. This byte stream contains all the information about the object. The serialization is commonly used for :-

>. To sent an object over a network.

>. To store an object in a database.

>. To save an objects state.

14). What is synchronization ?

Synchronization is a process of controlling access to shared resources by multiple threads. It also ensures that only one thread can access in a shared resource at a time. It ensure data security also.

15). Types of mechanisms for synchronization?

>. Synchronized methods :- It is a method that can only be executed by one thread at a time. When a thread calls a synchronized method , it puts a lock on the object that the method is associated with. So no other threads can call any synchronized method on the same object until the first thread release the lock.

>. synchronized statements:- It is an important concepts in multi-threading .It is a block of code that can only be executed by one thread at a time. When a thread entries a synchronized statement, it puts a lock on the object that the statement is associated with. So no other thread can enter any synchronized statement on the same object until the first thread realizes the lock.

16). What is thread in java?

Thread is a smaller unit of processing within a program. Each threads represents a separate flow of control and allowing multiple tasks to be performed simultaneously.

Eg:- if there is a kitchen. Each workers in the kitchen is threads. And we can consider kitchen as a program. Each workers (threads) are doing different works simultaneously. So each workers are thread.

17). Ways to achieve threads in java?

>Extend thread to class

>implementing runnable interface.

18). What is threading?

Threading is a technique of utilizing threads in a program by managing and coordinating multiple threads to achieve a specific task.

19). Threading vs multi-threading

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | Features | Threading | Multi\_threading | | Definition | This is the process of executing multiple threads within a single process. | This is the process of execution of multiple threads with in a process at the same time (concurrent execution). | | Nature | Used in single core system and multicore system | Especially designed for multi core system. | | Memory usage | Each threads within a process typically shares the same memory space | Multiple threads within a process share the same memory space. | | Example | Python threading module | Java’s thread class | |

20). What is java memory model?

JMM is a feature that describes how the java virtual machine manages memory. It also defines that how threads interact with memory and ensure that changes made by one threads are visible to others. In java heap is the memory area were storing the objects. And cache is the smallest and fastest memory that stores the recently accessed data.

21). What is JVM?

JVM stands for java virtual machine .which is a runtime environment that allows the java program to run on the different platforms. The function of JVM is converting every program code into byte code and after converting the bytecode to the machine specific code. That’s why java is platform independent.

The JVM is also responsible for the functions like garbage collection, memory management and security etc.

21). Why java is simple?

Java is easy to learn .Java is designed to easy to use ,because is to write codes, easy to debug, easy to compile, easy to learn than other programing languages.

23). Why Java is robust?

Java is robust because of the following features:-

>. strong memory management.

>. Exception handling .

>. Platform independence.

>. Automatic garbage collection.

>. Security features.

24). Why java is secure?

Java is a secured programing language because of the following reasons such as. :-

>. The java programs are run inside in a virtual machine which is known as sandbox. So java does not support explicit pointers.

>. Bytecode verification : Java code is compiled to byte code .this helps to ensure that only the valid codes is executing.

>. Automatic memory management: The garbage collector is automatically identifies the memory spaces in the heap that do not have any references and free upping the spaces .

25). What is JDK?

JDK stands for Java Development kit. Which is the software development kit for java developers. It consist of bundle of libraries ,tools ,Java runtime environment (JRE) and JDK.

26). What is JRE ?

JRE stands for Java Runtime Environment. It is a software package that provides the runtime environment for the java application. Which means it’s the package of software that provides the JVM, Class libraries and other necessary files to run the java application properly.

27).What is method overloading in java?

It defines that a class having one or more methods with same name but different arguments . The method is calling according to the similarities in parameters .this concepts is known as method over loading. it is often use to implement polymorphism in the object oriented programing.

28). What is method overriding?

The method over riding is a feature of java. When a sub class extends/implements the superclass/interface . The subclass provides a specific implementation of a method that is already defined in the superclass. This is called method overriding .

This allows the subclass to customize the behavior of the method defied in the superclass. It is often use to implement polymorphism in the object oriented programing language.

29). What are the features of OOPS?

**> Encapsulation :-**  Encapsulation is a programing concept that bundles the data and methods into a single unit like capsule. We can achieve encapsulation by use the access modifier private . when we declare the variable as private in the classes then it will only modifies using getter and setter methods. Through making the class private, restrict the direct access of the class.

**> Polymorphism** **:-** We can achieve polymorphism through method overriding and method overloading .

> **Abstraction** **:-**  It is the process of hiding the implementation of logic and showing only necessary details to the users. We can achieve abstraction by using interfaces/abstract classes and abstract methods.

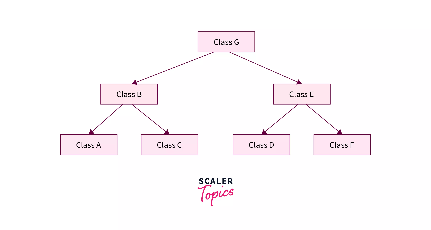
> **Inheritance** **:-** This is the process of inheriting the features of the parent class to the child class.

30). Types of inheritance?

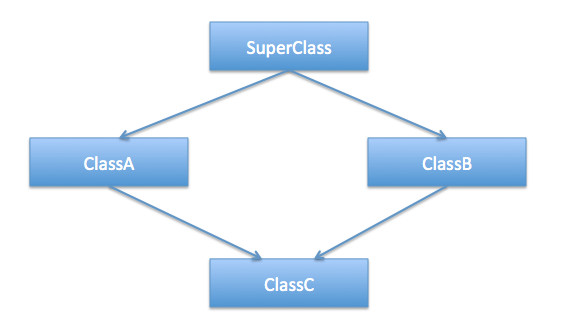
**> Single level inheritance** :- In this inheritance a child class inherits all the properties of only one parent class.

**> Multi level inheritance** :- its flow goes like class B inherits A,C inherits B, D inherits C. So the last class get all the propertied of above classes.

**> Hierarchial inheritance** :- Its flow is in the hierarchical structure.



**> Multiple inheritance** :- It is not a good practice and it doesn’t promote java. .because if there Is grand parent class having two child class but the two child class cannot becomes the parent of another child class .if that’s happen that’s leads to more complex and confusion.



**> Hybrid Inheritance** :- Hybrid inheritance is the combination of multiple and Hierarchical inheritance.so it is not possible in java .because if there Is grand parent class having two child class but the two child class cannot becomes the parent of another child class .if that’s happen that’s leads to more complex and confusion.

31). Main memory areas in java ?

->. Stack => A stack is a linear data structure that follows the LIFO principle and which is use to store function calls and variables.

->. Heap => Heap memory is used for the dynamic memory allocation and deallocation of java objects. Which storing objects created at runtime ,such as arrays , classes and collections. The JVM is automatically manages the memory allocation and deallocation for objects in heap.

->. Method Area => It is a memory block that stores the static variables runtime constants etc. when we declare a function in java as static it will keep in the method area.

->. Program Counter => It holds the address of the currently executing JVM instruction. The PC register is update each time a JVM instruction is executed.

->. Native method stack => This area holds the native methods information and is used when java code calls native codes.

32). What is completable future ?

The completable future is a class in java which is introduced in java 8 which is used for asynchronous programing in java . it is in the part of the java.util.concurrent .

33). What is the difference between == and equals() ?

🡪. == is used for comparing the primitive data types (int ,double, Boolean ..etc.)and object references .

🡪. Equals() is used to compare the content of two objects.

Eg:-

String s1 = new String("hello");

String s2 = new String("hello");

System.out.println(s1 == s2); // false (different memory locations)

System.out.println(s1.equals(s2)); // true (contents are equal)

Integer i1 = 10;

Integer i2 = 10;

System.out.println(i1 == i2); // true (autoboxing caches small integer values)

System.out.println(i1.equals(i2)); // true (contents are equal)

34). What Class loader in java?

In java , Class loader is a sub system of JVM. That is responsible for loading classes and interfaces as they are referenced by java program. It also ensures that necessary class files are loaded in to memory, so that java program can execute them.

There are several class loaders.

🡪. Bootstrap class loaders

🡪. Extension class loaders

🡪. System class loaders

🡪. Custom class loaders

35). What is deep copy?

The deep copy in java refers to creating a new object and then copying the content of the another object into it , recursively copying all nested objects . This ensures that the copied object and its contents are completely independent from the original objects. Which means the modifications made on the copied objects are not affects the original objects.

36). What is dependency injection?

It is a programing technique in which an object or functions receives other objects or functions that it requires. There are three types of dependency injections :-

🡪. Constructions injection

🡪. Setter injection

🡪. Method injection

36). What is the difference between comparator and comparable?

Both are used with streams in java for sorting elements.

🡪. Comparable is an interface .By implementing the comparable interface a class indicates that its instance have a natural order. And it provides a way to compare the instance of that class with one another.

The ***compareTo()*** is compares the current object with another .

🡪. Comparator :- When we need to define custom comparison logic for the objects we can use comparator.

37). What is IOC ?

The Inversion Of Control is a design principle in software engineering where the control flow of a program is inverted . The frame work or container takes the entire control and calls the necessary methods or functions based on the configuration.