

Learning from Failure: My First Java Mock Interview Experience



Q1. Can you give an intro yourself?

Ans: I have given my intro

Q2. Tell me what is Abstraction in Java and how would implement in Java

Ans. Abstraction is one of the principles of OOP. Abstraction we will use to hide implementation and I have given the example of a car where we will have a start button brake as abstraction and internal implementation is hidden and I told him we will achieve this using interfaces in Java.

Code

```
public Interface DisplayInterface{
void display();
}

public class DisplayImpl Implements DisplayInterface{
    void display(){
        System.out.print.out("this is display method")
    }
}

public class Client {
    Public Static Main(args[] str){
        DisplayImpl display = new DisplayImpl();
        display.display();
    }
}
```

Q3. Tell me the significance of Static keyword in Java.

Ans. My explanation static keyword whenever we will use it, it belongs to the class and if we will use it on variable we can directly get it using class name, if we will use it on method we can directly call it using class name because this method is the same for all instances.

Q4. Can you write a sample code for this and implement increment the count?

Code

```
public class StaticClass{
    static int number = 0;
    public static void getNumber(){
        return this.number;
    }

    public static void increment(int num){
        this.number += num;
    }
}

public class Client {
    Public Static Main(args[] str){
        StaticClass.incrementNum(3);
        StaticClass.incrementNum(3);
    }
}
```

how much it will return?

ans: I told if 6 which was correct because the static method called twice and every time I am calling the static method its value is final and adding 3 on it.

He told logically it's fine but compile this code there is some issue in the code.

Q5. Tell me the difference between Runnable and Callable.

Ans. Runnable and Callable both are interfaces. Runnable has the run method and Callable has call method. We will use Runnable when we have to run the task and Callable returns you the value after execute.

Q6. write the code using 2 threads; only thread 1 should print odd numbers and thread 2 will print even numbers.

Code

```
public class PrintNumberTask implements Runnable{  
    // Overwrite  
    public void run(num){  
        System.out.print(num);  
    }  
}
```

```
public class Client extends Thread {  
    Public Static Main(args[] str){  
        Thread t1 = new Thread();  
        for (int i = 1; i < 100; i += 2) {  
            PrintNumberTask task = new PrintNumberTask();  
            t1.start(task);  
        }  
    }  
}
```

```
Thread t2 = new Thread();  
for (int i = 0; i < 100; i += 2) {  
    PrintNumberTask task =  
        new PrintNumberTask();  
    t2.start(task);  
}
```

Mistake done by me, I should call the loop inside the PrintNumberTask or either like below: Using lambda function

```
public class Main {  
    public static void main(String[] args) {  
        Thread t1 = new Thread(() -> {  
            for (int i = 1; i < 100; i += 2) {  
                System.out.println("Odd: " + i);  
            }  
        });  
  
        Thread t2 = new Thread(() -> {  
            for (int i = 0; i < 100; i += 2) {  
                System.out.println("Even: " + i);  
            }  
        });  
  
        t1.start();  
        t2.start();  
    }  
}
```


Q7. Tell me the internal working of HashMap in Java.

Ans. HashMap saves the key-value pair and whenever it will store all the node in array buckets in the node it will save 4 things 1 s the hashCode, next key, next value and store next reference and it will provide the value by the key and it is not thread safe.

Q8. How HashMap generate hashCode?

Ans: I do not know that much detail.

Q9. Tell me the difference between Comparable and Comparator in Java

Ans. Comparable and Comparator both use for sorting whenever we use the custom natural order sorting we will use Comparable. It is an interface we will implement Comparable and implement compareTo function directly in the class and Comparator we will use when we need multiple sorting logic by passing the custom comparator.

Q10. Write the code for descending sort.

```
Collections.sort(numberList, (Integer num1, Integer num2) -> num2 - num1);
```

Q11. Write the code for streams which will return me odd numbers.

Ans. First I explained to him what is stream. We will use stream to use functional programming inside Java. By nature, it is lazy load whenever we have to use filter, map we will use stream and there is a terminal function whenever we will use it, it will return data else it is not return data it will return stream only that's why it called lazy loading.

```
numberList.stream().filter((int num) -> num % 2 != 0).forEach((item -> System.out.print(item)));
```

Q12. Tell me the usages of the keyword final, finally, and finalize.

Ans

Final -> Whenever we do not want to make any change in a variable we will use **final** whenever we apply final to the class we cannot create a subclass of that class.

finally -> we will use whenever we will use try-catch block and it will work always even you get error or not but there is one condition it will not work when the program terminates before reaching to this block.

Finalize -> is used for to tell the garbage collector before collecting the object in garbage it will check the object and normally we will not use finalize because it is unpredictable and causes performance issues.

Feedback given by Manivannan:

You are good in theory but theory does not matter in company practical is very important and you have to more practice on code because this is just a mock interview do not get demotivated keep practicing more and I am rejecting you. Keep practicing without IDE so you remember syntax.

Final Result -> Weak Reject

@GeekySanjay

Thank You



Contact Me

If you have any questions or suggestions regarding the video,
please feel free to reach out to me

on WhatsApp: **Sanjay Yadav Phone: 8310206130**

<https://www.linkedin.com/in/yadav-sanjay> | <https://www.youtube.com/@GeekySanjay>

