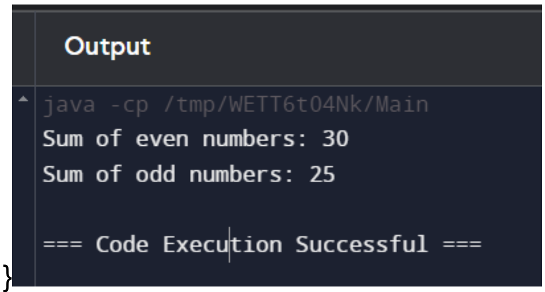
Generic method(01-08-2024)

write a java program to create generic method that take list of numbers and return the sum of all the even and odd number

```
import java.util.List;

public class Main {

    public static void main(String[] args) {

        List<Integer> numbers = List.of(1, 2, 3, 4, 5, 6, 7, 8, 9, 10);

        System.out.println("Sum of even numbers: " + sumEven(numbers));

        System.out.println("Sum of odd numbers: " + sumOdd(numbers));

    }

    public static <T extends Number> int sumEven(List<T> numbers) {

        int sum = 0;

        for (T number : numbers) {

            if (number.intValue() % 2 == 0) {

                sum += number.intValue();

            }

        }

        return sum;

    }

    public static <T extends Number> int sumOdd(List<T> numbers) {

        int sum = 0;

        for (T number : numbers) {

            if (number.intValue() % 2 != 0) {

                sum += number.intValue();

            }

        }

        return sum;

  }
```



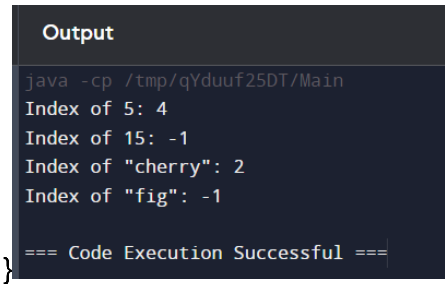
```
java -cp /tmp/WETT6t04Nk/Main
Sum of even numbers: 30
Sum of odd numbers: 25

=== Code Execution Successful ===
```

}

write a java program to create generic method that take list of any types and a target element it return the index if 1st occurrence of target element in the list return -1 if the target element cannot be found.

```
import java.util.List;

public class Main {

    public static void main(String[] args) {

        List<Integer> numbers = List.of(1, 2, 3, 4, 5, 6, 7, 8, 9, 10);

        System.out.println("Index of 5: " + indexOf(numbers, 5));

        System.out.println("Index of 15: " + indexOf(numbers, 15));


        List<String> strings = List.of("apple", "banana", "cherry", "date", "elderberr");

        System.out.println("Index of \"cherry\": " + indexOf(strings, "cherry"));

        System.out.println("Index of \"fig\": " + indexOf(strings, "fig"));

    }

    public static <T> int indexOf(List<T> list, T target) {

        for (int i = 0; i < list.size(); i++) {

            if (list.get(i).equals(target)) {

                return i;

            }

        }

        return -1;

    }

}
```

}

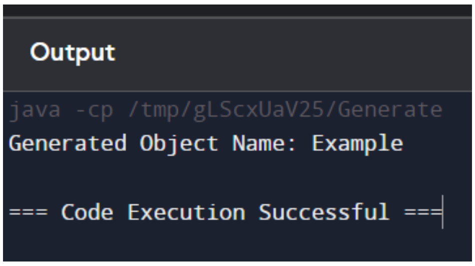


Output

```
java -cp /tmp/qYduuf25DT/Main
Index of 5: 4
Index of 15: -1
Index of "cherry": 2
Index of "fig": -1

=== Code Execution Successful ===
```

create generate class with another generate for the object of the generate class.

```
public class Generate {
    private String name;
    public Generate(String name) {
        this.name = name;
    }
    public class GeneratedObject {
        private String name;
        public GeneratedObject(String name) {
            this.name = name;
        }
        public void printName() {
            System.out.println("Generated Object Name: " + name);
        }
    }
    public GeneratedObject generateObject() {
        return new GeneratedObject(name);
    }
    public static void main(String[] args) {
        Generate generate = new Generate("Example");
        Generate.GeneratedObject generatedObject = generate.generateObject();
        generatedObject.printName();
    }
}
```



```
java -cp /tmp/gLScxUaV25/Generate
Generated Object Name: Example

=== Code Execution Successful ===
```