DAY 23:

ASSIGNMENT 8:

Task 8: Generics and Type Safety

Create a generic Pair class that holds two objects of different types, and write a method to return a reversed version of the pair."

```
public class Triple<T1, T2, T3> {

   private T1 first;

   private T2 second;

   private T3 third;


   // Constructor to initialize the Triple

   public Triple(T1 first, T2 second, T3 third) {

      this.first = first;

      this.second = second;

      this.third = third;

   }


   // Getters for the elements

   public T1 getFirst() {

      return first;

   }
```

```
    public T2 getSecond() {
        return second;
    }

    public T3 getThird() {
        return third;
    }

    // Method to return a new Triple with the elements rotated to the right
    public Triple<T3, T1, T2> rotate() {
        return new Triple<>(third, first, second);
    }

    // Main method to test the Triple class
    public static void main(String[] args) {
        // Create a Triple object with a String, an Integer, and a Double
        Triple<String, Integer, Double> originalTriple = new Triple<>("Hello", 123, 45.67);

        // Print the original Triple
        System.out.println("Original Triple: (" + originalTriple.getFirst() + ", " + originalTriple.getSecond() + ", " + originalTriple.getThird() + ")");

        // Get the rotated Triple
        Triple<Double, String, Integer> rotatedTriple = originalTriple.rotate();

        // Print the rotated Triple
        System.out.println("Rotated Triple: (" + rotatedTriple.getFirst() + ", " + rotatedTriple.getSecond() + ", " + rotatedTriple.getThird() + ")");
    }
}
```

Explanation:

1. *Generic Type Parameters*: The Triple class is defined with three type parameters, T1, T2, and T3, representing the types of the first, second, and third elements, respectively.

2. *Constructor*: The constructor initializes the first, second, and third fields with the provided values.

3. *Getters*: The getFirst, getSecond, and getThird methods return the first, second, and third elements of the triple, respectively.

4. *Rotate Method*: The rotate method creates and returns a new Triple object with the types and values of the elements shifted to the right.

5. *Main Method*: The main method demonstrates the usage of the Triple class by creating a triple, printing it, rotating it, and printing the rotated triple.