## ASSIGNMENT – COURIER MANAGEMENT SYSTEM

## Coding

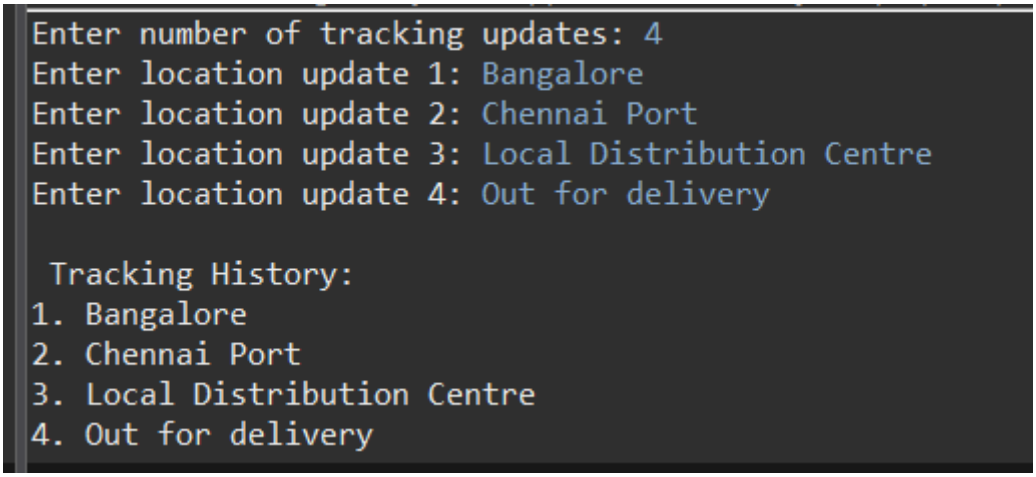
## Task 3: Arrays and Data Structures

**7. Create an array to store the tracking history of a parcel, where each entry represents a location update.**

```
import java.util.*;
public class TrackingHistory {
        public static void main(String[] args) {
                Scanner sc = new Scanner(System.in);
      System.out.print("Enter number of tracking updates: ");
      int n = sc.nextInt();
      sc.nextLine();
      String[] trackingHistory = new String[n];
      for (int i = 0; i < n; i++) {
         System.out.print("Enter location update " + (i + 1) + ": ");
         trackingHistory[i] = sc.nextLine();
      }
      System.out.println("\n Tracking History:");
      for (int i = 0; i < n; i++) {
         System.out.println((i + 1) + ". " + trackingHistory[i]);
      }

      sc.close();

        }
}
```




**8. Implement a method to find the nearest available courier for a new order using an array of couriers.**

```
import java.util.*;


public class NearestCourier {
        public static int findNearestCourier(int[] courierDistances) {
      int nearestIndex = 0;
      int minDistance = courierDistances[0];


      for (int i = 1; i < courierDistances.length; i++) {
         if (courierDistances[i] < minDistance) {
            minDistance = courierDistances[i];
            nearestIndex = i;
         }
      }


      return nearestIndex;
   }
   public static void main(String[] args) {
      Scanner scanner = new Scanner(System.in);
```

```
        System.out.print("Enter number of couriers: ");
        int n = scanner.nextInt();
        int[] courierDistances = new int[n];


        for (int i = 0; i < n; i++) {
            System.out.print("Enter distance of courier: ");
            courierDistances[i] = scanner.nextInt();
        }

        int nearest = findNearestCourier(courierDistances);

        System.out.println("\nNearest available courier is at index " + nearest +
                           " with distance " + courierDistances[nearest] + " km.");

        scanner.close();
    }
}
```