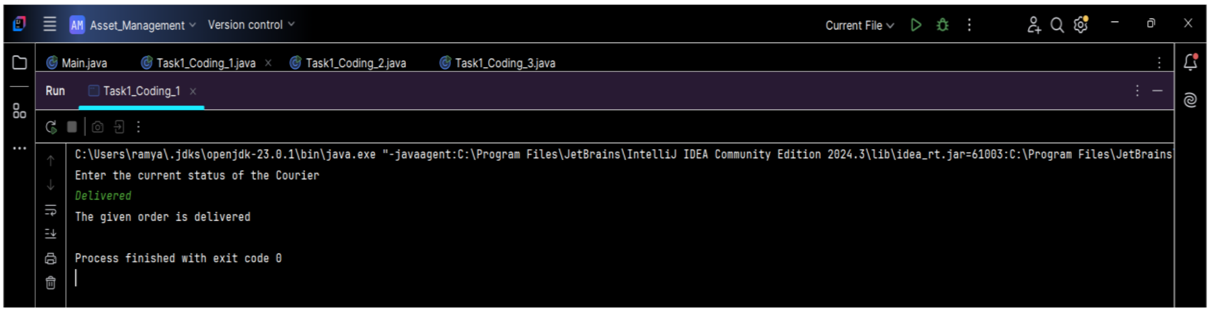
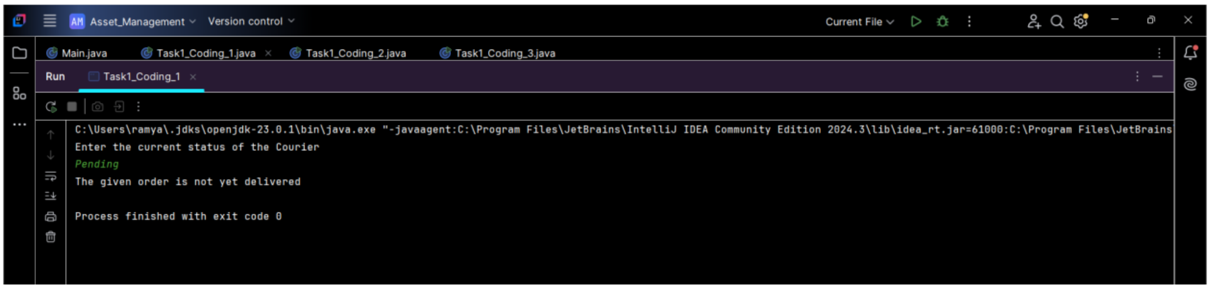
Task 1: Control Flow Statements

1. Write a program that checks whether a given order is delivered or not based on its status (e.g., "Processing," "Delivered," "Cancelled"). Use if-else statements for this.

```
import java.util.Scanner;
public class Task1_Coding_1
{

   public static void statuscheck(String status)
   {
      if(status.equals("Delivered") || status.equals("delivered") ||
status.equals("DELIVERED"))
         System.out.println("The given order is delivered");
      else
         System.out.println("The given order is not yet delivered");
   }
   public static void main(String[] args)
   {
           Scanner sc=new Scanner(System.in);
           System.out.println("Enter the current status of the Courier");
           String status=sc.nextLine();
           statuscheck(status);

}

}
```



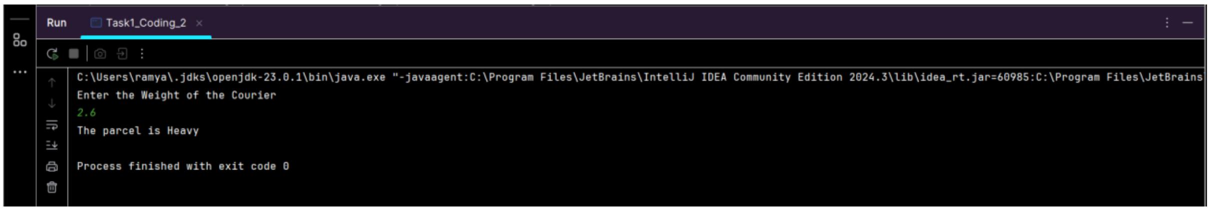
```
C:\Users\ramya\.jdks\openjdk-23.0.1\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2024.3\lib\idea_rt.jar=61003:C:\Program Files\JetBrains
Enter the current status of the Courier
Delivered
The given order is delivered

Process finished with exit code 0
```



```
C:\Users\ramya\.jdks\openjdk-23.0.1\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2024.3\lib\idea_rt.jar=61000:C:\Program Files\JetBrains
Enter the current status of the Courier
Pending
The given order is not yet delivered

Process finished with exit code 0
```

2. Implement a switch-case statement to categorize parcels based on their weight into "Light," "Medium," or "Heavy."

```
import java.util.Scanner;

public class Task1_Coding_2
{
   public static void Weight_check(double weight)
   {
      int specified_number;
      if(weight<1.5)
         specified_number=1;
      else if(weight<2.5 && weight>1.5)
         specified_number=2;
      else
         specified_number=3;

      switch (specified_number) {
         case 1:
            System.out.println("The parcel is Light");
            break;
         case 2:
            System.out.println("The parcel is  Medium");
            break;
         case 3:
            System.out.println("The parcel is Heavy");
            break;
         default:
            System.out.println("Entered Weight is Invalid");
      }

   }
   public static void main(String[] args)
   {
      Scanner sc = new Scanner(System.in);
      System.out.println("Enter the Weight of the Courier");
      double weight=sc.nextDouble();
      Weight_check(weight);
   }
}
```

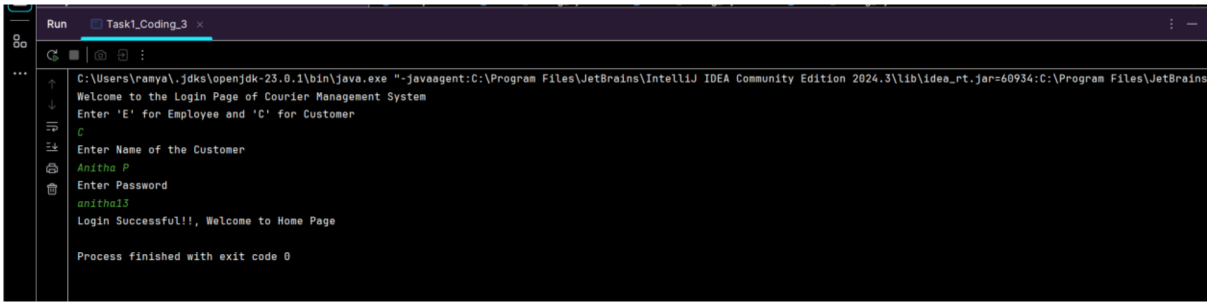


```
C:\Users\ramya\.jdks\openjdk-23.0.1\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2024.3\lib\idea_rt.jar=60985:C:\Program Files\JetBrains
Enter the Weight of the Courier
2.6
The parcel is Heavy

Process finished with exit code 0
```

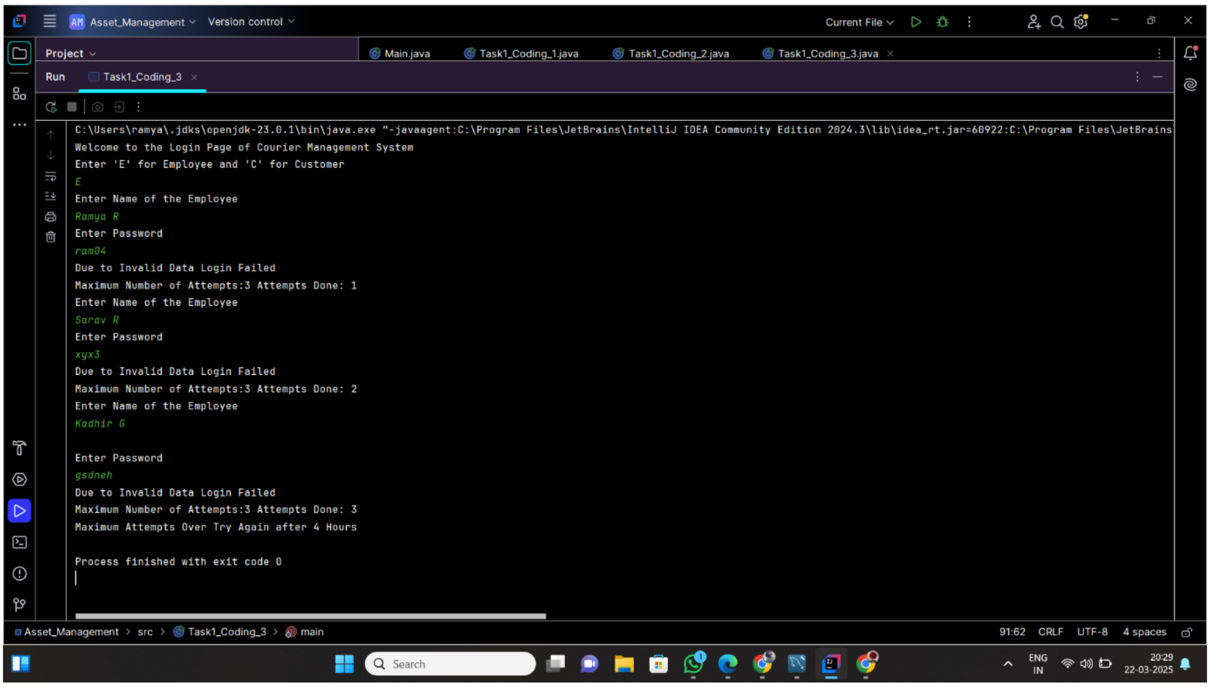
3. Implement User Authentication 1. Create a login system for employees and customers using Java control flow statements.

```
import java.util.HashMap;
import java.util.Scanner;

public class Task1_Coding_3
{

    public static void main(String[] args)
    {
        HashMap<String, String> Customers = new HashMap<>();
        HashMap<String, String> employees = new HashMap<>();


        // Datas stored in Database
        employees.put("Divya Shree H", "divya45");
        employees.put("Meena T", "meena@13");
        employees.put("Manoj K", "kmanoj76");
        employees.put("Suresh B", "suresh*9");
        employees.put("Monika S", "smoni9");
        Customers.put("Arun Kumar", "ak47");
        Customers.put("Karthik S", "karthi870");
        Customers.put("Sundar C", "sundar*c01");
        Customers.put("Anitha P", "anitha13");
        Customers.put("Revathi R", "revathisk2");


        // Scanner class to get to know the user is customer or employee
        Scanner sc = new Scanner(System.in);
        System.out.println("Welcome to the Login Page of Courier Management System");
        System.out.println(("Enter 'E' for Employee and 'C' for Customer"));
        char input = sc.next().charAt(0);


        // To access Number of attempts done
        int attempts = 0;
        String ename, epassword, cname, cpassword;
        // USed to check Authentication status
        boolean AuthenticationSuccess = false;
        // Code if user is employee
        if (input == 'E')
        {
          while (attempts < 3 && !AuthenticationSuccess)
          {
            System.out.println("Enter Name of the Employee");
            sc.nextLine();
            ename = sc.nextLine();
            System.out.println("Enter Password");
            epassword = sc.nextLine();
            if (input == 'E')
            {
              AuthenticationSuccess = Validate(ename, epassword, employees);
            }
            if (AuthenticationSuccess)
            {
              System.out.println("Login Successful!!, Welcome to Home Page");
            } else
            {
              attempts++;
              System.out.println("Due to Invalid Data Login Failed");
              System.out.println("Maximum Number of Attempts:3 Attempts Done: " +
attempts);
            }

          }
          //After 3 attempts Authentication not get success
          if (!AuthenticationSuccess)
          {
            System.out.println("Maximum Attempts Over Try Again after 4 Hours");
          }

        }
        // Code if user is customer
        else
        {
          while (attempts < 3 && !AuthenticationSuccess)
          {

            System.out.println("Enter Name of the Customer");
            sc.nextLine();
            cname = sc.nextLine();
```

```
                System.out.println("Enter Password");
                cpassword = sc.nextLine();
                if (input == 'C')
                {
                    AuthenticationSuccess = Validate(cname, cpassword, Customers);
                }
                if (AuthenticationSuccess)
                {
                    System.out.println("Login Successful!!, Welcome to Home Page");
                } else
                {
                    attempts++;
                    System.out.println("Due to Invalid Data Login Failed");
                    System.out.println("Maximum Number of Attempts:3 Attempts Done: " +
attempts);
                }

            }
            //After 3 attempts Authentication not get success
            if (!AuthenticationSuccess)
            {
                System.out.println("Maximum Attempts Over Try Again after 4 Hours");
            }
        }
}

    public static boolean Validate(String username, String password, HashMap<String, String>
users)
    {
        return users.containsKey(username) && users.get(username).equals(password);
    }

}
```



```
C:\Users\ramya\.jdks\openjdk-23.0.1\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2024.3\lib\idea_rt.jar=60934:C:\Program Files\JetBrains
Welcome to the Login Page of Courier Management System
Enter 'E' for Employee and 'C' for Customer
C
Enter Name of the Customer
Anitha P
Enter Password
anitha13
Login Successful!!, Welcome to Home Page

Process finished with exit code 0
```



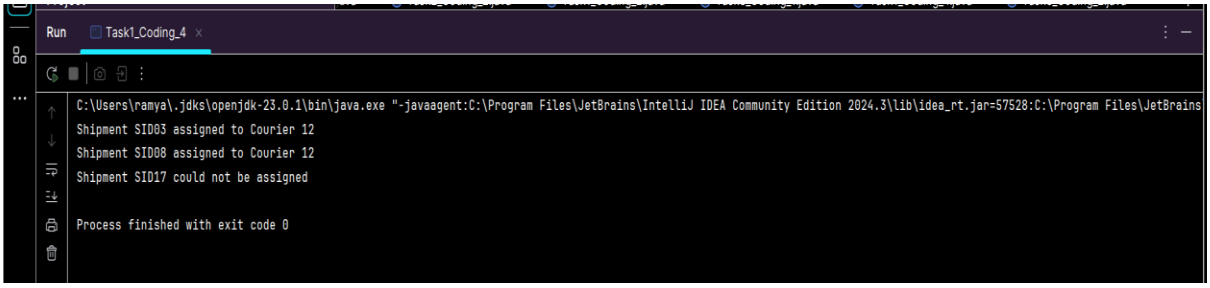
```
C:\Users\ramya\.jdks\openjdk-23.0.1\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2024.3\lib\idea_rt.jar=60922:C:\Program Files\JetBrains
Welcome to the Login Page of Courier Management System
Enter 'E' for Employee and 'C' for Customer
E
Enter Name of the Employee
Ramya R
Enter Password
ram04
Due to Invalid Data Login Failed
Maximum Number of Attempts:3 Attempts Done: 1
Enter Name of the Employee
Sarav R
Enter Password
xyx3
Due to Invalid Data Login Failed
Maximum Number of Attempts:3 Attempts Done: 2
Enter Name of the Employee
Kadhir G

Enter Password
gsdneh
Due to Invalid Data Login Failed
Maximum Number of Attempts:3 Attempts Done: 3
Maximum Attempts Over Try Again after 4 Hours

Process finished with exit code 0
```

Asset_Management > src > Task1_Coding_3 > main
91:62 CRLF UTF-8 4 spaces
Search
ENG IN
20:29
22-03-2025

4. Implement Courier Assignment Logic 1. Develop a mechanism to assign couriers to shipments based on predefined criteria (e.g., proximity, load capacity) using loops.

```
class Courier {
    int CourierID;
    double distance;
    int capacity;

    Courier( int CourierID, double distance, int capacity){
        this.CourierID = CourierID;
        this.distance = distance;
        this.capacity = capacity;
    }
}
class Shipment {
    String id;
    int requiredCapacity;

    public Shipment(String id, int requiredCapacity) {
        this.id = id;
        this.requiredCapacity = requiredCapacity;
    }
}
```

```java
public class Task1_Coding_4 {
    public static void main(String[] args) {
        Courier[] courier = {
                new Courier(12, 2.56, 23),
                new Courier(9, 3.4, 8),
                new Courier(6, 9.865, 10)

        };
        Shipment[] shipment = {
                new Shipment("SID03", 5),
                new Shipment("SID08", 7),
                new Shipment("SID17", 15)
        };
        for (Shipment s : shipment) {
            boolean assigned = false;

            for (Courier c : courier) {
                if (c.capacity >= s.requiredCapacity) {
                    System.out.println("Shipment " + s.id + " assigned to Courier " + c.CourierID);
                    c.capacity -= s.requiredCapacity; // Update courier capacity
                    assigned = true;
                    break;  // Move to the next shipment after assigning
                }
            }
            if (!assigned) {
                System.out.println("Shipment " + s.id + " could not be assigned");
            }

        }
    }
}
```

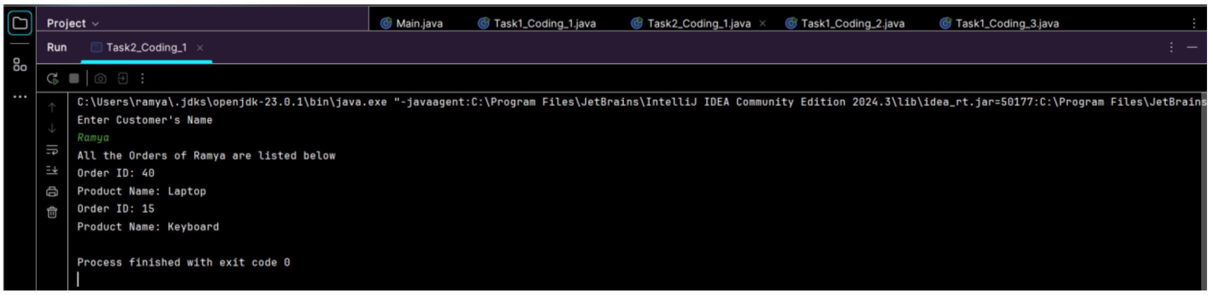


```
C:\Users\ramya\.jdks\openjdk-23.0.1\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2024.3\lib\idea_rt.jar=57528:C:\Program Files\JetBrains
Shipment SID03 assigned to Courier 12
Shipment SID08 assigned to Courier 12
Shipment SID17 could not be assigned

Process finished with exit code 0
```

Task 2: Loops and Iteration

5. Write a Java program that uses a for loop to display all the orders for a specific customer.

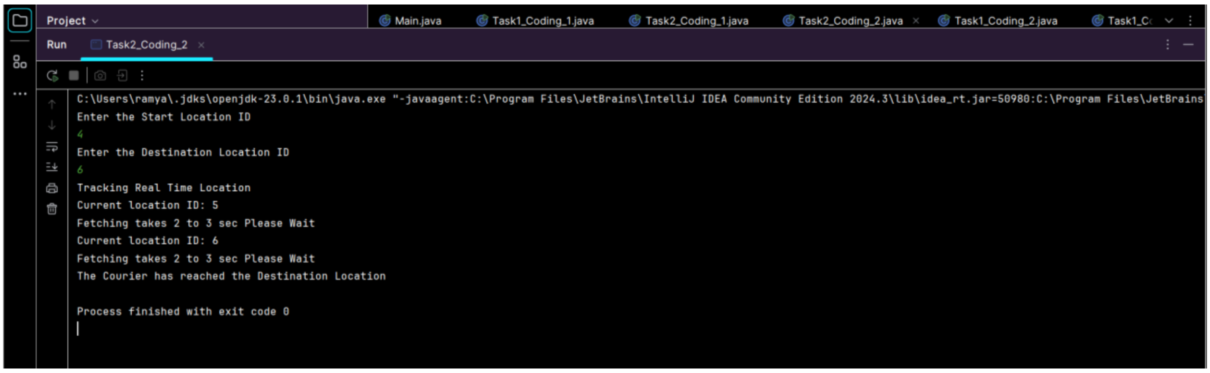
```
import java.util.*;
class Order
{
   int orderID;
   String CustomerName;
   String Product;
   Order(int orderID,String CustomerName, String Product)
   {
      this.orderID=orderID;
      this.CustomerName=CustomerName;
      this.Product=Product;
   }
}
public class Task2_Coding_1 {

   public static void main(String[] args)
   {
      Order[] order ={
            new Order(40,"Ramya","Laptop"),
            new Order(86,"Yadav","Projector"),
            new Order(13,"Kailash","Smart Watch"),
            new Order(15,"Ramya","Keyboard"),
            new Order(17,"Sarav","MacBook"),
            new Order(50,"Sarav","Tripod")
      };
      Scanner sc=new Scanner(System.in);
      System.out.println("Enter Customer's Name");
      String cname=sc.next();
      System.out.println("All the Orders of "+cname+" are listed below");
      for(Order o:order)
      {
         if(o.CustomerName.equalsIgnoreCase(cname)) {
            System.out.println("Order ID: " + o.orderID);
            System.out.println("Product Name: "+o.Product);
         }
      }

  }
}
```



```
C:\Users\ramya\.jdks\openjdk-23.0.1\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2024.3\lib\idea_rt.jar=50177:C:\Program Files\JetBrains
Enter Customer's Name
Ramya
All the Orders of Ramya are listed below
Order ID: 40
Product Name: Laptop
Order ID: 15
Product Name: Keyboard

Process finished with exit code 0
```

6. Implement a while loop to track the real-time location of a courier until it reaches its destination.

```
import java.util.Scanner;

public class Task2_Coding_2
{
    public static void main(String[] args)
    {
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter the Start Location ID");
        int s_id=sc.nextInt();
        System.out.println("Enter the Destination Location ID");
        int d_id=sc.nextInt();
        System.out.println("Tracking Real Time Location");
        while(s_id != d_id)
        {
            try {
                Thread.sleep(3000);
            }

            catch (InterruptedException e) {
                System.out.println("Interrupted Exception occurred");
            }
            if(s_id<d_id){
                s_id++;
            }else  {
                s_id--;
            }
                System.out.println("Current location ID: " + s_id);
                System.out.println("Fetching takes 2 to 3 sec Please Wait");


        }
             System.out.println("The Courier has reached the Destination Location");
    }
}
```

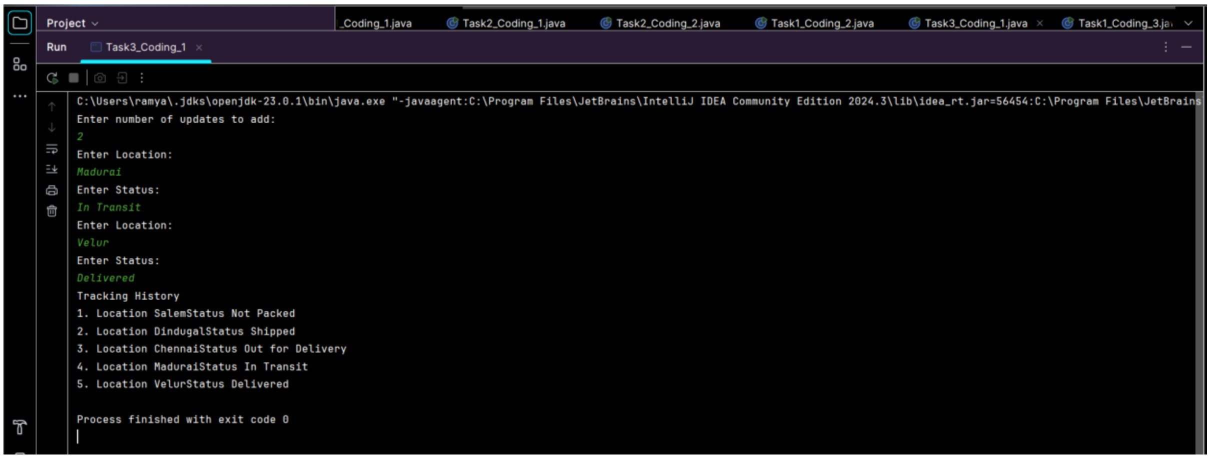


```
C:\Users\ramya\.jdks\openjdk-23.0.1\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2024.3\lib\idea_rt.jar=50980:C:\Program Files\JetBrains
Enter the Start Location ID
4
Enter the Destination Location ID
6
Tracking Real Time Location
Current location ID: 5
Fetching takes 2 to 3 sec Please Wait
Current location ID: 6
Fetching takes 2 to 3 sec Please Wait
The Courier has reached the Destination Location

Process finished with exit code 0
```

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

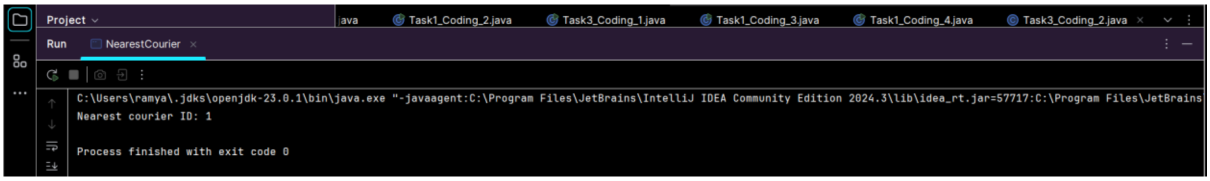
```
import java.util.Scanner;

public class Task3_Coding_1 {
    String location;
    String status;

    Task3_Coding_1(String location, String status) {
        this.location = location;
        this.status = status;
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);


        Task3_Coding_1[] tracking = new Task3_Coding_1[15];
        tracking[0] = new Task3_Coding_1("Salem", "Not Packed");
        tracking[1] = new Task3_Coding_1("Dindugal", "Shipped");
        tracking[2] = new Task3_Coding_1("Chennai", "Out for Delivery");
        int count=3;
        System.out.println("Enter number of updates to add:");
        int updates = sc.nextInt();
        sc.nextLine();

        // Loop to get Entry from User
        for (int i = 0; i < updates; i++) {

            System.out.println("Enter Location:");
            String location = sc.nextLine();
            System.out.println("Enter Status:");
            String status = sc.nextLine();
            tracking[count] = new Task3_Coding_1(location, status);
            count++;
        }
        //Loop to display the updated tracking History

        System.out.println("Tracking History");
        for (int i = 0; i < count; i++) {
            System.out.println((i+1)+". Location "+tracking[i].location+"Status
"+tracking[i].status);
        }

    }
}
```



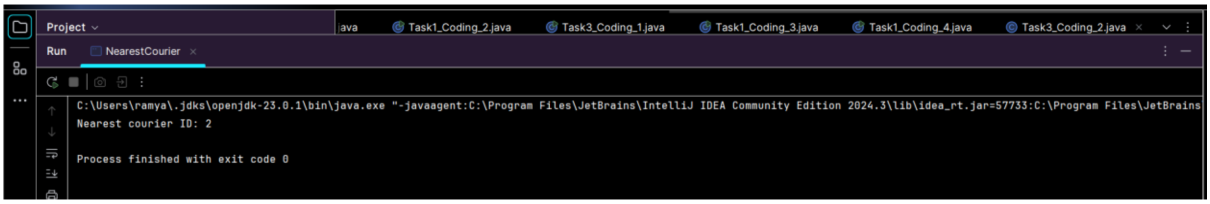
```
C:\Users\ramya\.jdks\openjdk-23.0.1\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2024.3\lib\idea_rt.jar=56454:C:\Program Files\JetBrains
Enter number of updates to add:
2
Enter Location:
Madurai
Enter Status:
In Transit
Enter Location:
Velur
Enter Status:
Delivered
Tracking History
1. Location SalemStatus Not Packed
2. Location DindugalStatus Shipped
3. Location ChennaiStatus Out for Delivery
4. Location MaduraiStatus In Transit
5. Location VelurStatus Delivered

Process finished with exit code 0
```

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

```
public class Task3_Coding_2 {

    int id;
    double distance; // Distance from a fixed origin
    boolean available;

    Task3_Coding_2(int id, double distance, boolean available) {
        this.id = id;
        this.distance = distance;
        this.available = available;
    }


  public class NearestCourier {
    public static Task3_Coding_2 findNearest(Task3_Coding_2[] couriers, double
orderDistance) {
        Task3_Coding_2 nearest = null;
        double minDistanceDiff = Double.MAX_VALUE;

        if (couriers == null || couriers.length == 0) {
          return null;
        }

        for (Task3_Coding_2 c : couriers) {
          if (c.available) {
            double distanceDiff = Math.abs(c.distance - orderDistance); // Absolute distance
difference
            if (distanceDiff < minDistanceDiff) {
              minDistanceDiff = distanceDiff;
              nearest = c;
            }
          }
        }
        return nearest;
    }

    public static void main(String[] args) {
        Task3_Coding_2[] couriers = {
            new Task3_Coding_2(1, 10.0, true),
            new Task3_Coding_2(2, 11.0, true),
            new Task3_Coding_2(3, 15.0, false),
            new Task3_Coding_2(4, 9.0, true)
        };

        Task3_Coding_2 nearest = findNearest(couriers, 10.5);

        if (nearest != null) {
          System.out.println("Nearest courier ID: " + nearest.id);
        } else {
          System.out.println("No available couriers.");
        }
    }
  }
}
```



```
C:\Users\ramya\.jdks\openjdk-23.0.1\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2024.3\lib\idea_rt.jar=57717:C:\Program Files\JetBrains
Nearest courier ID: 1

Process finished with exit code 0
```



```
C:\Users\ramya\.jdks\openjdk-23.0.1\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2024.3\lib\idea_rt.jar=57733:C:\Program Files\JetBrains
Nearest courier ID: 2

Process finished with exit code 0
```