Contents

TITLE

STUDENT ID :

STUDENT NAME :

MODULE CODE : COMP 20014

MODULE NAME : Adavanced Programming

TASK 2 AND 3

In this task I have developed a client side and server side in which client will send some data to the server and server will perform some operations and return back the result to be displayed on the client screen. As both task 2 and 3 were very small and linked with one another so, I have done it in one code.

Client Code Java

```
package main;

import java.io.IOException;
import java.io.PrintStream;
```

```
import java.net.InetAddress;
import java.net.Socket;
import java.net.UnknownHostException;
import java.sql.SQLException;
import java.util.NoSuchElementException;
import java.util.Scanner;

public class client {

	public static void main(String[] args) throws UnknownHostException,
IOException {

		int custId, amountSpent, pointsEarned, baisaEarned;
		Scanner s = new Scanner(System.in);
		Socket ss = new Socket("127.0.0.1", 1204); // socket toconnect to server
		Scanner din = new Scanner(ss.getInputStream()); // to get the values
back from server
		PrintStream dout = new PrintStream(ss.getOutputStream()); // to send
data to server
		custId = 1;
		amountSpent = 500;
		dout.println(custId);
		dout.println(amountSpent);
		pointsEarned = din.nextInt();
		baisaEarned = din.nextInt();
		System.out.println("Points Earned : " + pointsEarned);
		System.out.println("Baisa Earned : " + baisaEarned);

	}

}
```

Server Code Java

```
package main;

import java.io.IOException;
import java.io.PrintStream;
import java.net.ServerSocket;
import java.net.Socket;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.Scanner;

public class server {

	public static void main(String[] args) throws IOException, SQLException,
ClassNotFoundException {
		int custId, amountSpent, pointsEarned, baisaEarned;
		ServerSocket serverSocket = new ServerSocket(1204); // to connect to
client
```

```
            Socket s = serverSocket.accept(); // to accept incoming data
            Scanner din = new Scanner(s.getInputStream());// to get the data
            PrintStream dout = new PrintStream(s.getOutputStream()); // to send back
data
            custId = din.nextInt();
            amountSpent = din.nextInt();
            pointsEarned = amountSpent * 50;
            baisaEarned = pointsEarned * 2;
            dout.println(pointsEarned);
            dout.println(baisaEarned);

      }
}
```

TASK 4

In this task I have created a database in mySql using xampp server and created a table.After that I linked my java application using JDBC connector to the database . Now user can add customer id and amount and then the program will check if that customer is already present in the table then update the points earned and baisa earned accordingly and if that customer is not present in the table then create a new one.

Client CODE JAVA

```
import java.io.IOException;
import java.io.PrintStream;
import java.net.Socket;
import java.util.Scanner;

public class client {

      public static void main(String[] args) throws IOException {
            // TODO Auto-generated method stub
            int custId,amountSpent,pointsEarned,baisaEarned,choice;
            Scanner s = new Scanner(System.in);
            Socket ss = new Socket("127.0.0.1",1204);
            Scanner din = new Scanner(ss.getInputStream());
            PrintStream dout = new PrintStream(ss.getOutputStream());
            do {
            System.out.println("Enter choice : ");
            System.out.println("1:customer ");
            System.out.println("2:Quit");
            choice = s.nextInt();

            if(choice == 1) {
```

```
            System.out.print("Please enter customer ID: ");
        custId = s.nextInt();
        System.out.print("Please enter amount spent: ");
        amountSpent = s.nextInt();
        dout.println(custId);
        dout.println(amountSpent);
        pointsEarned = din.nextInt();
        baisaEarned = din.nextInt();
        System.out.println("pointsEarned : "+pointsEarned);
        System.out.println("pointsEarned : "+baisaEarned);


            }
            }while(choice == 1);

      }

}
```

Server Code Java

```
import java.io.DataInputStream;
import java.io.DataOutputStream;
import java.io.IOException;
import java.io.PrintStream;
import java.net.ServerSocket;
import java.net.Socket;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.Scanner;

public class server {

      public static void main(String[] args) throws SQLException,
ClassNotFoundException {
            // TODO Auto-generated method stub
            int
custId,amountSpent,pointsEarned,baisaEarned,choice,newAmount,oldPoints,newPoints,oldB
aisa,newBaisa;
             try {
                  ServerSocket serverSocket = new ServerSocket(1204);
                  Socket s = serverSocket.accept();
                  Scanner din = new Scanner(s.getInputStream());
                  PrintStream dout = new PrintStream(s.getOutputStream());
                  custId = din.nextInt();
                  amountSpent = din.nextInt();
                  pointsEarned = amountSpent*50;
                  baisaEarned = pointsEarned*2;
```

```
                    Class.forName("com.mysql.jdbc.Driver");
                     Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/test?serverTimezone=UTC",
"root", "");

                    Statement stat = con.createStatement();
                           String sqlSelect = "select * from testassignment where
custId = "+custId;
                           System.out.println(custId);
                           ResultSet rs = stat.executeQuery(sqlSelect);
                           if (rs.next()) {
                            newAmount = rs.getInt("amountSpent")+amountSpent;
                            System.out.println(newAmount);
                            oldPoints = rs.getInt("pointsEarned");
                            newPoints = pointsEarned+oldPoints;
                            oldBaisa = rs.getInt("baisaEarned");
                            newBaisa = oldBaisa+baisaEarned;
                            PreparedStatement update = con.prepareStatement("UPDATE
testassignment SET amountSpent = ?, pointsEarned = ?, baisaEarned = ? WHERE custId =
?");
                             update.setInt(1, newAmount);
                             update.setInt(2, newPoints);
                             update.setInt(3, newBaisa);
                             update.setInt(4, custId);
                             update.execute();


                          con.close();
                          dout.println(newPoints);
                              dout.println(newBaisa);
                           }

                           else if (!rs.next()) {
                                  PreparedStatement insert =
con.prepareStatement("insert into  testassignment values ( ?, ?, ?, ?)");
                                  insert.setInt(1, custId);
                                  insert.setInt(2, amountSpent);
                                  insert.setInt(3, pointsEarned);
                                  insert.setInt(4,baisaEarned);
                                  insert.executeUpdate();
                                  dout.println(pointsEarned);
                          dout.println(baisaEarned);
                                  con.close();
                           }


               }
               catch (IOException ioex){
                   System.out.println("Unable to attach to port!");
                   System.exit(1);
               }

      }

}
```

TASK 5

In this task I have created a treemap and a main menu through which user can select various options like adding value to tree , searching the tree or sorting the tree.The sorting is done through comparator.

Code TreeMap Java

```
package treeMap;

import java.util.ArrayList;
import java.util.Comparator;
import java.util.List;
import java.util.Map;
import java.util.Scanner;
import java.util.SortedSet;
import java.util.TreeMap;
import java.util.TreeSet;

public class treeMapClass {
	static int choice;

	public static void main(String[] args) {
		// TODO Auto-generated method stub
		int custId, amountSpent, oldAmount, oldPoints, oldBaisa, newPoints,
newAmount, newBaisa;
		Scanner input = new Scanner(System.in);
		List<Integer> list1 = new ArrayList<Integer>(); // list to handle
customer data
		List<Integer> list2 = new ArrayList<Integer>();
		TreeMap<Integer, Integer> map = new TreeMap<Integer, Integer>(); // tree
map storing custId and the
		System.out.println("Enter Choice : "); // main menu
		System.out.println("1: Add to Tree ");
		System.out.println("2: Sort Tree");
		System.out.println("3: Search Tree");
		System.out.println("4: quit");

		choice = input.nextInt();
		do { // data related to cust.

			switch (choice) {
			case 1:

				System.out.println("Enter custId : ");
				custId = input.nextInt();
				System.out.println("Enter amount spent: ");
				amountSpent = input.nextInt();
```

```
                        if (map.containsKey(custId)) { // check if cust already
present ina tree
                              oldAmount = map.get(custId);
                              newAmount = oldAmount + amountSpent;
                              map.replace(custId, newAmount);
                        } else if (!map.containsKey(custId)) {
                              map.put(custId, amountSpent); // add custId and data
to tree map
                        }
                        break;

                  case 2:
                        System.out.println(entriesSortedByValues(map)); // call
sorted method
                        break;
                  case 3:
                        System.out.println("Enter custId : ");
                        custId = input.nextInt();
                        if (map.containsKey(custId)) {
                              System.out.println(map.get(custId));
                        } else if (!map.containsKey(custId)) {
                              System.out.println("this customer id is not present
in the tree map");
                        }
                        break;

                  case 4:
                        System.exit(0);

                  }

                  System.out.println("Enter Choice : "); // main menu
                  System.out.println("1: Add to Tree ");
                  System.out.println("2: Sort Tree");
                  System.out.println("3: Search Tree");
                  System.out.println("4: quit");

                  choice = input.nextInt();

            } while (choice != 4);
      }

      static <K, V extends Comparable<? super V>> SortedSet<Map.Entry<K, V>>
entriesSortedByValues(Map<K, V> map) {
            SortedSet<Map.Entry<K, V>> sortedEntries = new TreeSet<Map.Entry<K,
V>>(new Comparator<Map.Entry<K, V>>() {
                  @Override
                  public int compare(Map.Entry<K, V> e1, Map.Entry<K, V> e2) {
                        int res = e1.getValue().compareTo(e2.getValue());
                        return res != 0 ? res : 1;
                  }
```

```
            });
            sortedEntries.addAll(map.entrySet());
            return sortedEntries;
        }

}
```