OOPM:-PRACTICAL 2

Name: Vivian Vijay Ludrick
Branch: SE Comps-A
Batch: C
Roll No.: 9914

Q.1) Write a program to create a class Student with data ‘name, city and age’ along with method printData to display the data. Create the two objects s1, s2 to declare and access the values.

```
class Student {
  private int id;
  private String name;
  private String city;
  private int age;

  // constructor to set the student data
  Student(int id, String name, int age, String city) {
    this.id = id;// assigns the local values to the class variables
    this.name = name;
    this.age = age;
    this.city = city;
  }

  // prints the student data
  void printData() {
    System.out.println("Student" + id + ":\n\tName: " + name +
"\n\tAge: " + age + "\n\tCity: " + city +
"\n_____________________");
  }
}// end of class Student

class StudentDetails {
  public static void main(String[] args) {

    // instantiating the objects of class Student and passing the
values to the constructors
    Student s1 = new Student(1, "Pratyay", 18, "Mumbai");
    Student s2 = new Student(2, "Shwen", 18, "Vasai");
    // yes i know vasai is not a city

    // print the data passed through the constructor
```

```
    System.out.println("\n_____________________");
    s1.printData();
    s2.printData();

  }
}
```

OUTPUT :

```
viraj@LAPTOP-EO8CTJ7K MINGW64
$ java StudentDetails

--------------------
Student1:
        Name: Pratyay
        Age: 18
        City: Mumbai
--------------------
Student2:
        Name: Shwen
        Age: 18
        City: Vasai
--------------------
```

Q.2) Write a program to create a class Student2 along with two method getData(),printData() to get the value through argument and display the data in printData. Create the two objects s1 ,s2 to declare and access the values from class STtest

```
class Student2 {
  private int id;
  private String name;
  private String city;
  private int age;

  // constructor to set the student data
  Student2(int id, String name, int age, String city) {
    this.id = id;// assigns the local values to the class variables
    this.name = name;
    this.age = age;
    this.city = city;
  }

  public int getId() {
    return id;
  }

  public String getName() {
    return name;
  }

  public int getAge() {
    return age;
  }

  public String getCity() {
    return city;
  }
}// end of class Student2

class STtest {
  private static int id;
  private static String name;
  private static String city;
  private static int age;

  static void setData(Student2 stud){
```

```
    id = stud.getId();
    name = stud.getName();
    age = stud.getAge();
    city = stud.getCity();
  }

  //prints the data to the user. used static as only static methods
can be referenced in other static methods.
  static void printData(){
    System.out.println("Student" + id + ":\n\tName: " + name +
"\n\tAge: " + age + "\n\tCity: " + city +
"\n_____________________");
  }

  public static void main(String[] args) {
    // instantiating the objects of class Student and passing the
values to the constructors
    Student2 s1 = new Student2(1, "Pratyay", 18, "Mumbai");
    Student2 s2 = new Student2(2, "Shwen", 18, "Vasai");
    // yes i know vasai is not a city

    System.out.println("\n_____________________");
    setData(s1);//passes the data of the student
    printData();// print the data passed through the constructor
    setData(s2);
    printData();
  }
}
```

OUTPUT :

```
viraj@LAPTOP-EO8CTJ7K MINGW64
$ java STtest

--------------------
Student1:
        Name: Pratyay
        Age: 18
        City: Mumbai
--------------------
Student2:
        Name: Shwen
        Age: 18
        City: Vasai
--------------------
```

Q.3) WAP using parameterized constructor with two parameters id and name. While creating the objects obj1 and obj2 passed two arguments so that this constructor gets invoked after creation of obj1 and obj2

```
class Object{
  private int id;
  private String name;

  //constructor
  Object(int id, String name){
    this.id = id;
    this.name = name;
  }

  //printing the object data passed through the constructor
  void printData(){
    System.out.println("Object" + id + ":-\tName: " + name +
"\n_____________________");
  }
}

class ObjectCreation{
  public static void main(String[] args) {
    Object obj1 = new Object(1, "first object");
    Object obj2 = new Object(2, "second object");

    System.out.println("\n_____________________");
    obj1.printData();
    obj2.printData();
  }

}
```

OUTPUT :

```
viraj@LAPTOP-EO8CTJ7K MINGW64 /c/Vivian/submissions
$ java ObjectCreation

--------------------
Object1:-       Name: first object
--------------------
Object2:-       Name: second object
--------------------
```

Q.4) WAP to demonstrate the working of a banking-system ,where we deposit and withdraw amount from our account. Creating an Account class which has deposit() and withdraw() methods.

```
import java.util.Scanner;

class Account {
  // object states
  private long accountNumber;
  private String accountHolderName;
  private String password;
  private double balance;
  boolean flagRightAccount = true;

  Scanner sc = new Scanner(System.in);

  // constructor to define the object data
  Account(long accNo, String name, double amount, String password)
{
    this.accountNumber = accNo;
    this.accountHolderName = name;
    this.balance = amount;
    this.password = password;
  }

  // verifies the account number
  boolean verifyAccountNumber(String input) {
    try {
      if (Long.parseLong(input) == this.accountNumber) {
        return true;
      } else {
        return false;
      }
    } catch (Exception e) {
      return false;
    }
  }

  // verifies the account name
  boolean verifyAccountHolderName(String input) {
    if (input.equals(this.accountHolderName)) {
      return true;
    }
```

```
    return false;
  }

  // verifies the password
  boolean verifyPassword(String input) {
    if (input.equals(this.password)) {
      return true;
    }
    return false;
  }

  // checks whether the account is right
  void checkAccount() {
    while (this.flagRightAccount) {
      // prints account details
      System.out.println("\n Please verify the account details to
proceed\n\tAccount Holder Name: "
          + this.getAccountHolderName() + "\n\tAccount Number: " +
this.getAccountNumber() + "\n\tBalance: "
          + this.getBalance());

System.out.println("-------------------------------------------------
-------------------------");
      System.out.print("\nWould you like to proceed?(yes /
no(default)) ");

      String proceed = sc.nextLine();// right account?

System.out.println("-------------------------------------------------
-------------------------");
      if (proceed.equalsIgnoreCase("yes")) {
        // account is correct
        this.flagRightAccount = false;
        operation();// goes to make an operation on the account
        return;
      } else {
        System.out.print("Would you like to re-enter your details?
(yes / no(default)):\t");// enter details again?
        String retry = sc.nextLine();

System.out.println("-------------------------------------------------
-------------------------");
```

```
        // reenter your details.
        if (retry.equalsIgnoreCase("yes")) {
          Bank.needflagDetails = true;
          Bank.accountExists = false;
          Bank.passwordMatches = false;// reenter account details
          Bank.verifyAccount();
        } else {

          System.out.println("--x--x--End of program--x--x--");
          System.exit(0);
          // return; //i have no idea why return doesn't work
        }
      }
    }
  }

  // tried recursion here
  // operation on the account
  void operation() {
    System.out.println(
        "Enter the mode of
operation:\n\t1.Withdraw(with)\n\t2.Deposit(dep)\n\t3.Balance(bal)
\n\t4.Change Password(pass)\n\t5.Go Back(back) \n\t6.Cancel
Operation(c)");
    String input = sc.nextLine();// operation input

System.out.println("-----------------------------------------------
-------------------------");
    switch (input) {
      case "with":
        withdraw();
        System.out.println("The account currently has Rs." +
getBalance() + "\\-");
        break;
      case "dep":
        deposit();
        System.out.println("The account currently has Rs." +
getBalance() + "\\-");
        break;
      case "bal":
        System.out.println("The account currently has Rs." +
getBalance() + "\\-");
```

```
        break;
      case "pass":
        setPassword();
        break;
      case "back":
        // goes to account
        this.flagRightAccount = true;
        checkAccount();
        return;
      case "c":
        // exit
        return;
      default:
        System.out.println("Error: Invalid input");
    }

System.out.println("-------------------------------------------------
-------------------------");
    // make another operation
    System.out.print("Would you like to make another
operation?(yes / no(default)):\t");
    String again = sc.nextLine();

System.out.println("-------------------------------------------------
-------------------------");
    if (again.equalsIgnoreCase("yes")) {
      operation();// recalls the operation again
    } else {
      this.flagRightAccount = true;
      checkAccount();// goes one step back
    }
  }

  // changes the password if the requirements are met
  public void setPassword() {
    System.out.print("Enter the previous password:\t");
    String previousPassword = sc.nextLine();
    System.out.print("Enter the new password:\t");
    String newPassword1 = sc.nextLine();
    System.out.print("Re-enter the new password:\t");
    String newPassword2 = sc.nextLine();
```

```
    // used .equals cause '==' doesn't work on strings
    if (previousPassword.equals(this.password)) {
      if (newPassword1.equals(newPassword2)) {
        this.password = newPassword1;
        System.out.println("The new password is: " +
this.password);
      } else {
        System.out.println("The new passwords dont match");
      }
    } else {
      System.out.println("The previous password is incorrect");
    }
  }

  // withdraw money from bank
  void withdraw() {
    System.out.print("Enter the amount to withdraw:\t");
    // makes sure that the input has double datatype
    while (!sc.hasNextDouble()) {
      System.out.print("Error: Invalid input. Please enter a valid
amount:\t");
      sc.nextDouble(); // Clear the invalid input
    }
    double withdrawnAmount = sc.nextDouble();
    sc.nextLine();// consumes the leftover newline

    // checks whether the amount already exists in the account or
not
    if (withdrawnAmount > 0 && withdrawnAmount > balance) {
      System.out
          .println("Error: Your withdrawing amount is more than
your current balance amount. Operation cannot proceed");
      return;
    } else {
      balance -= withdrawnAmount;
      System.out.println("You have withdrawn Rs." + withdrawnAmount
+ "\\- from your bank account");
      return;
    }
  }

  // deposit money in bank
```

```
  void deposit() {
    System.out.print("Enter the amount to deposit:\t");
    // makes sure that the input has double datatype
    while (!sc.hasNextDouble()) {
      System.out.println("Error: Invalid input. Please enter a
valid amount:");
      sc.nextDouble(); // Clear the invalid input
    }
    double depositedAmount = sc.nextDouble();
    sc.nextLine();
    if (depositedAmount < 1) {
      System.out.println("Error: Minimum Amount Allowed:\tRs.1");
      return;
    } else {
      balance += depositedAmount;
      System.out.println("You have deposited Rs." + depositedAmount
+ "\\- in your bank account");
      return;
    }
  }

  // getter balance
  public double getBalance() {
    return balance;
  }

  // getter account name
  public String getAccountHolderName() {
    return accountHolderName;
  }

  // getter account number
  public long getAccountNumber() {
    return accountNumber;
  }

}// end of class Account

class Bank {

  static boolean needflagDetails = true;// for repetative
crendentials input
```

```
  // for account verfication
  static boolean accountExists = false;
  static boolean passwordMatches = false;

  static Account[] ac = new Account[3];// initialised outside so
that both functions can access it
  static int i = 0;// current account

  // verifies whether the account exist and the user can be logged
in
  static void verifyAccount() {
    Scanner sc = new Scanner(System.in);

    // repetative checking of the credentials
    while (needflagDetails) {
      // input account details
      System.out.print("Enter the account holder's name/ account
number:\t");
      String name = sc.nextLine();
      System.out.print("Enter the account holder's password:\t");
      String pass = sc.nextLine();

System.out.println("------------------------------------------------
-------------------------");

      // checks each account to find a match
      for (i = 0; i < ac.length; i++) {
        if (ac[i].verifyAccountHolderName(name) ||
ac[i].verifyAccountNumber(name)) {
          if (ac[i].verifyPassword(pass)) {
            accountExists = true;
            passwordMatches = true;
            break;
          } else {
            accountExists = true;// only account exists
          }
        }
      }

      // output based on the entered credentials
      if (accountExists && passwordMatches) {
```

```
        needflagDetails = false;
        System.out.println("User Logged in!");
        ac[i].checkAccount();// logs the user into its account
        return;
      } else if (accountExists && !passwordMatches) {
        accountExists = false;
        System.out.println("Incorrect password");
      } else {
        System.out.println("Invalid username.");
      }

System.out.println("-----------------------------------------------
-------------------------");
      // whether to ask for the detials again or not
      if (needflagDetails) {
        System.out.print("Would you like to retry?(yes /
no(default)):\t");
        String retryInput = sc.nextLine();
        if (retryInput.equalsIgnoreCase("yes")) {
          needflagDetails = true;
        } else {
          needflagDetails = false;
        }
      }

System.out.println("-----------------------------------------------
-------------------------");

    }
  }

  public static void main(String[] args) {
    // initialising the accounts
    ac[0] = new Account(1l, "Shaun Mendes", 100000, "pass1");
    ac[1] = new Account(2l, "Mark Lopes", 100000, "pass2");
    ac[2] = new Account(3l, "Jonathan Gomes", 100000, "pass3");

    // this method is used to call all the operations. The methods
of the objects
    // call other method until the user doesn't want to continue
with the operation
    verifyAccount();
```

```
  }
}// it works
```

OUTPUT :

```
viraj@LAPTOP-EO8CTJ7K MINGW64 /c/Vivian/submissions
$ java Bank
Enter the account holder's name/ account number:        2
Enter the account holder's password:    pass2
--------------------------------------------------------------------------
User Logged in!

 Please verify the account details to proceed
        Account Holder Name: Mark Lopes
        Account Number: 2
        Balance: 100000.0
--------------------------------------------------------------------------

Would you like to proceed?(yes / no(default)) yes
--------------------------------------------------------------------------
Enter the mode of operation:
        1.Withdraw(with)
        2.Deposit(dep)
        3.Balance(bal)
        4.Change Password(pass)
        5.Go Back(back)
        6.Cancel Operation(c)
with
--------------------------------------------------------------------------
Enter the amount to withdraw:   123
You have withdrawn Rs.123.0\- from your bank account
The account currently has Rs.99877.0\-
--------------------------------------------------------------------------
Would you like to make another operation?(yes / no(default)):   yes
--------------------------------------------------------------------------
Enter the mode of operation:
        1.Withdraw(with)
        2.Deposit(dep)
```

```
        3.Balance(bal)
        4.Change Password(pass)
        5.Go Back(back)
        6.Cancel Operation(c)
dep
--------------------------------------------------------------------------
Enter the amount to deposit:    345
You have deposited Rs.345.0\- in your bank account
The account currently has Rs.100222.0\-
--------------------------------------------------------------------------
Would you like to make another operation?(yes / no(default)):   yes
--------------------------------------------------------------------------
Enter the mode of operation:
        1.Withdraw(with)
        2.Deposit(dep)
        3.Balance(bal)
        4.Change Password(pass)
        5.Go Back(back)
        6.Cancel Operation(c)
bal
--------------------------------------------------------------------------
The account currently has Rs.100222.0\-
--------------------------------------------------------------------------
Would you like to make another operation?(yes / no(default)):   no
--------------------------------------------------------------------------

 Please verify the account details to proceed
        Account Holder Name: Mark Lopes
        Account Number: 2
        Balance: 100222.0
--------------------------------------------------------------------------

Would you like to proceed?(yes / no(default)) no
--------------------------------------------------------------------------
```

```
 --------------------------------------------------------------------------
 Would you like to re-enter your details? (yes / no(default)):   no
 --------------------------------------------------------------------------
 --x--x--End of program--x--x--
```

Q. 5) Write a program to print the area of a rectangle by creating a class named 'Area' taking the values of its length and breadth as parameters of its constructor and having a method named 'returnArea' which returns the area of the rectangle. Length and breadth of rectangle are entered through keyboard.

```
import java.util.Scanner;//importing the Scanner class

class Area {
  private double length;
  private double breadth;

  // constructor to input the dimensions of the rectangle
  Area(double len, double bre) {
    this.length = len;
    this.breadth = bre;
  }

  // returns the area of rectangle
  double returnArea() {
    return this.length * this.breadth;
  }

}

class Rectangle {
  public static void main(String[] args) {
    double length;
    double breadth;
    double area;

    Scanner sc = new Scanner(System.in);// creating an object ot use scanner
class
    System.out.print("Enter the length of the rectangle:\t");
    length = sc.nextDouble();
    sc.nextLine();// consuming the newLine character
    System.out.print("Enter the breadth of the rectangle:\t");
    breadth = sc.nextDouble();
    sc.nextLine();// consuming the newLine character
    Area rect1 = new Area(length, breadth);// creating an object of class area

    area = rect1.returnArea();// stores the area of the object rect1

    System.out.println("The area of the rectangle is " + area);
  }
}
```

OUTPUT :

```
viraj@LAPTOP-EO8CTJ7K MINGW64 /c/Vivian/submissions
$ java Rectangle
Enter the length of the rectangle:      45
Enter the breadth of the rectangle:     30
The area of the rectangle is 1350.0
```