Date 19/01/2024
Page

1. Develop a Java program to create an abstract class named Shape that contains two integers and an empty method named printArea(). Provide three classes named Rectangle, Triangle and Circle such that each one of the class extends the class Shape. Each one of the class contain the method printArea() that prints the area of the given shape.

```
abstract class Shape {
    protected int length;
    protected int width;

    public Shape (int length, int width) {
        this.length = length;
        this.width = width;
    }
    public abstract void printArea();
}

class Rectangle extends Shape {
    public Rectangle (int length, int width) {
        super (length, width);
    }
    public void printArea() {
        int area = length * width;
        System.out.println ("Rectangle Area: " + area);
    }
}
```

Date __/__/__
Page

```
class Triangle extends Shape {
    public Triangle(int length, int width) {
        super(length, width);
    }
    public void printArea() {
        double area = 0.5 * length * width;
        System.out.println("Triangle Area: " + area);
    }
}

class Circle extends Shape {
    public Circle(int radius) {
        super(radius, 0);
    }

    Public void printArea() {
        double area = Math.PI * length * length;
        System.out.println("Circle Area" + area);
    }
}

public void printArea() {
    double area = Math
}

public class Main {
    public static void main(String[] args) {
        Rectangle rectangle = new Rectangle(5, 10);
        rectangle.printArea();
        Triangle triangle = new Triangle(3, 8);
        triangle.printArea();
        Circle circle = new Circle(4);
        Circle.printArea();
    }
}
```

Date __/__/__
Page _____

output:

Rectangle Area: 50
Area of Triangle: 17
Area of circle: 50.

2 Develop a Java program to create a class Bank that maintains two kinds of account for its customers, one called savings account and the other current account. The savings account provides compound interest and withdrawal facilities but no cheque book facility. The current account provides cheque book facility but no interest. Current account holders should also maintain a minimum balance and if the balance falls below this level a service charge is imposed.

Create a class Account that stores customer name, Account number and type of account. From this derive the classes Cur-acct and Sav-acct to make them more specific to their requirements. Include the necessary methods in order to achieve the following tasks.

a) Accept deposit from customer and update the balance.

b) Display the balance

c) Compute and deposit interest

d) Permit withdrawal and update the balance.

Check for the minimum balance, impose penalty if necessary and update the balance

Date
Page

```
import java.util.Scanner;

class Account {
    protected String customerName;
    protected long accountNumber;
    protected String accountType;
    protected double balance;

    public Account(String customerName, long accountNumber, String accountType, double balance) {
        this.customerName = customerName;
        this.accountNumber = accountNumber;
        this.accountType = accountType;
        this.balance = balance;
    }

    public void displayBalance() {
        System.out.println("Account Number: " + accountNumber);
        System.out.println("Customer Name: " + customerName);
        System.out.println("Account Type: " + accountType);
        System.out.println("Balance: $" + balance);
    }

    public void deposit(double amount) {
        balance += amount;
        System.out.println("Deposit of $" + amount + " Successful.");
        displayBalance();
    }

    public void withdraw(double amount) {
        if (amount <= balance) {
            balance -= amount;
            System.out.println("Withdrawal of $" + amount +
```

```
"Successful.");
} else {
  System.out.println("Insufficient funds. Withdrawal failed.");
}
}
displayBalance() {
}
}
```

count

```
class CurrAcct extends Account {
  private double minimumBalance = 1000;
  private double serviceCharge = 50;

  public CurrAcct(String customerName, long accountNumber, double balance) {
    super(customerName, accountNumber, "current", balance);

  public void withdraw(double amount) {
  if (amount <= balance - minimumBalance) {
  balance -= amount;
  System.out.println("Withdrawal of $" + amount + "Successful"); }
  else {
  System.out.println("Insufficient funds. Withdrawal failed. Service charge of $" + serviceCharge + " imposed");
  balance -= serviceCharge;
  }
  displayBalance();
  }
}
```

nt Number);
rrName)
tType);
t + "
amount +

Date __/__/__
Page _____

```
class SavAcct extends Account {
    private double interestRate = 0.05;

    public SavAcct(String customerName, long
accountNumber, double balance) {
        super(customerName, accountName, "Savings",
balance);
    }

    public void computeInterest() {
        double interest = balance * interestRate;
        balance += interest;
        System.out.println("Interest of $" + interest +
" credited");
        displayBalance();
    }
}

public class exp2 {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        SavAcct savingsAccount = new SavAcct
("Farhan", 123456789, 5000);
        SavingsAccount.displayBalance();
        savingsAccount.deposit(1000);
        savingAccount.computeInterest();
        savingAccount.withdraw(2000);

        CurrAcct currentAccount = new CurrAcct("Farhan",
987432l, 1500);
        currentAccount.displayBalance();
        currentAccount.deposit(500);
```

Date __/__/__
Page _____

```
currentAccount.withdraw(2000);
scanner.close();
}
}
```

Output:

```
Account Number: 12345678 9.
Customer Name: Farhan.
Account Type: Savings.
Balance: $5000.0.
Deposit of $1000.0 successful.
Account Number: 12345678 9.
Customer Name: Farhan.
Account Type: Savings.
Balance: $6000.0.
Interest of $3000.0 Credited.
Account Number: 12345678 9
Customer Name: Farhan.
Account Type: Savings.
Balance: $6300.00.
Withdrawal of $2000.0 successful.
```

19.01.24