

## LAB-3

classmate \_\_\_\_\_

Date \_\_\_\_\_

Page \_\_\_\_\_

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

=>

abstract class Shape {

    int side1;

    int side2;

    Shape(int side1, int side2) {

        this.side1 = side1;

        this.side2 = side2;

    }

    abstract void printArea();

class Rectangle extends Shape {

    Rectangle(int L, int b) {

        super(L, b);

    }

    void printArea() {

        int area = side1 \* side2;

        System.out.println

(“Area of rectangle is:” +

area);

3

```
class Triangle extends Shape {
```

```
    Triangle (int a, int b)
```

```
        super (b, b),
```

```
}
```

```
    void paintArea () {
```

```
        int area = 0.5 * side1 * side2;
```

```
        System.out.println ("The area of
```

```
        the triangle is : " + area)
```

```
}
```

```
class Circle extends Shape {
```

```
    Circle (int r) {
```

```
        super (r, 0);
```

```
}
```

```
    void paintArea () {
```

```
        double Area = 3.14 * side1 * side2;
```

```
        System.out.println ("The area of
```

```
        the circle is : " + (area));
```

```
}
```

```
class Run {
```

```
    public static void main (String [] args) {
```

```
        Rectangle r = new Rectangle (2, 3);
```

```
        Triangle t = new Triangle (2, 3);
```

```
        Circle c = new Circle (7);
```

```
        r.paintArea ();
```

```
        t.paintArea ();
```

```
        c.paintArea ();
```

```
?
```

```
3
```

Output:

Area of rectangle is 6

Area of Triangle is 3.0

Area of Circle is 153.8613

D

2. Develop a Java program to create a class Bank that maintain two kinds of accounts for its customers, one called Saving account ~~provide~~ and the other current accounts.

The Savings account provides compound interest and withdrawal facilities but no cheque book facility. The current account provides check book facility but no interest. Current account holders should also maintain a minimum balance. If the balance falls below this level, a service charge is imposed.

Create a class Account that stores customer account number & type of account. From this derive the class cur-act & sav-act to make them more specific to their requirements. Include the necessary methods in order to achieve the following tasks:

- a) Accept deposit from customer & update the balance
- b) display the balance
- c) Compute & deposit interest
- d) Permit withdrawal & update the balance.  
→ Check for minimum balance, impose penalty if necessary & update the balance.

⇒

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

```
class account {
```

```
    String name;
```

```
    int accno;
```

```
    String type;
```

```
    double balance;
```

```
    account (String name, int accno, String type,  
            double balance) {
```

```
        this.name = name;
```

```
        this.accno = accno;
```

```
        this.type = type;
```

```
        this.balance = balance;
```

{

```
    void deposit (double amount) {
```

```
        balance += amount;
```

{

```
    void withdraw (double amount) {
```

```
        if ((balance - amount) >= 0)
```

```
            balance -= amount;
```

~~else~~

```
            System.out.println ("Insufficient  
                                balance");
```

{

```
    void display () {
```

```
        System.out.println ("name : " + name +
```

```
                            "accno : " + accno + "type : " + type +
```

```
                            "balance : " + balance);
```

{

{

class SavAcct extends Account {  
private static double rate = 5;

SavAcct (String name, int accno, double balance) {

super (name, accno, "saving balance");

}

void interest() {

{ balance += balance \* rate / 100;

System.out.println ("balance: ",  
balance);

}

3

class CurAcct extends Account {  
private double minBal = 500;  
private double serviceCharges = 50;

CurAcct (String name, int accno,  
double balance) {

super (name, accno, "current", balance);

}

3

void checkMin() {

if (balance < minBal) {

System.out.println ("balance is less than min balance");

balance -= serviceCharges;

System.out.println ("balance + balance);

}

3

3

```
class accountMain {
```

```
    public static void main(String ar[]) {
```

```
        Scanner s = new Scanner(System.in);
```

```
        System.out.println("Enter the name:");
```

```
        String name = s.next();
```

```
        System.out.printIn("Enter the type  
        (current/savings);")
```

```
        String type = s.next();
```

```
        System.out.printIn("Enter account no.");
```

```
        int accno = s.nextInt();
```

```
        System.out.printIn("Enter the initial  
balance: ");
```

```
        double balance = s.nextDouble();
```

```
        int ch;
```

```
        double amount1, amount2;
```

```
        account acc = new account(name,  
        accno, type, balance);
```

```
        current ac = new curAct(name,  
        accno, balance);
```

```
        current ca = new curAct(name,  
        accno, balance);
```

```
        while (true) {
```

```
            if (acc.type.equals("savings"))
```

```
                System.out.printIn(
```

```
System.out.println ("In Menu\n 1. deposit  
 2. withdraw\n 3. compute interest  
 4. display");
```

```
System.out.println ("Enter the choice");
```

```
ch = s.nextInt();
```

```
switch (ch) {
```

```
    case 1 : s.o.p ("Enter amount");  
    amount1 = s.nextInt();  
    sa.deposit (amount1);  
    break;
```

```
    case 2 : System.out.println ("Enter amount");  
    amount2 = s.nextInt();  
    sa.deposit (amount2);  
    break;
```

~~```
    case 3 : sa.interest ();  
    break;
```~~~~```
    case 4 : sa.display ();  
    break;
```~~~~```
    case 5 : System.exit (0);  
    break;
```~~~~```
    default : s.o.p ("invalid input");  
    break;
```~~

3

3

else {

SOP("In Menu In 1. deposit 12. withdraw  
So 3. display.

SOP ("enter the choice : ");

ch = S.nextInt();

switch(ch) {

case 1 : SOP("Enter amount ");

amount 1 = S.nextInt();

ca.deposit(amount1);

break;

case 2 : SOP("Enter the amount");

amount 2 = S.nextInt();

ca.withdraw(amount2);

ca.checkBalance();

break;

case 3 : ca.display();

break;

case 4 : System.exit(0);

default : SOP("In valid input");

break;

3

?

?