SAIPRAVEEN MARNI
1BM19CS138

28/10/20

LAB-3

① Write a java program to find roots of quadratic equation

```
import java.util.scanner;
import static java.lang.math.*;
class quadratic
{
  public static void main(String args[])
  {
    quadratic obj = new quadratic();
    Scanner sc = new Scanner(System.in);
System.out.println("Enter the value of a::");
  float a = sc.nextFloat();
System.out.println("Enter the value of b::");
  float b = sc.nextFloat();
System.out.println("Enter the value of c::");
  float c = sc.nextFloat();
      if(a==0)
    {
```

```
        System.out.println ("Invalid");
        return;
    }
    float d = b*b - 4*a*c;
    float sqrt_val = (float)Math.sqrt (abs(d));
    float root1 = (-b+sqrt_val) / (2*a);
    float root2 = (-b+sqrt_val) / (2*a);
    if (d==0)
{
System.out.println ("Roots are real and equal :: "+root1);
}
    else if (d>0)
{
    System.out.println ("Roots are real and different \n");
    System.out.println (root1 + "\n" + root2);
}
    else
    {
      System.out.println("Roots are complex \n");
    System.out.print(-b/(2*a)+"+i"+sqrt_val/(2*a)+
                    "\n" + -b/(2*a)+"-i"+sqrt_val/(2*a));
    }
  }
}
```

* ALGORITHM :-

Step 1 : START

Step 2 : Input the value of a, b, c.

Step 3 : Calculate d = b*b - 4*a*c

Step 4 : If (d<0) Display "Roots are imaginary", calculate r1 = (-b+sqrt-val)/2a and r2 = (-b -sqrt-val)/2a . else if (d=0) Display "Roots are Equal" then calculate r1 = r2 = (-b/2a).

Step 5 : print r1 and r2.

Step 6 : END.

→ Expected Output ←

Enter the value of a :: 3
Enter the value of b :: 2
Enter the value of c :: 2
Roots are complex
-0.33333334 + 0.74535 6
-0.33333334 - 0.74535 6

09/10/20

LAB-4

SAI PRAVEEN MARENI
1BM19CS138

Develop a JAVA program to create a class student with members usn, name, an array credits and an array marks. Include methods to accept and display details and a method to calculate SGPA of a student.

```
import java.util.Scanner;

class student
{
  String USN;
  String name;
  int n;
double SGPA = 0;
 int Credits = 0;
Scanner in = new Scanner(System.in);
 void Details()
{
System.out.println("Enter the USN of the student");
   USN = in.nextLine();
System.out.println("Enter the name of the student");
   name = in.nextLine();
System.out.println("Enter number of subjects");
    n = in.nextInt();
 int credits = new int[n];
double marks = new double[n];
System.out.println("Enter details of the subjects:");
  for(int i=0; i<n; i++)
{
```

```
System.out.println("Enter credits alloted to subjects" +(i+1));
  credits[i] = in.nextInt();
System.out.println("Enter marks in subject" +(i+1));
  marks[i] = in.nextInt();
Calculate(credits[i], marks[i], i);
}

}  void Calculate(int credit, double mark, int j)
{
  Credits = Credits + credit;
if (mark >= 90 && mark <= 100)
    SGPA = SGPA + (10* credit);
else if (mark >= 80 && mark <= 89)
    SGPA = SGPA + (9* credit);
else if (mark >=70 && mark <= 79).
    SGPA = SGPA + (8* credit);
else if (mark >= 60 && mark <= 69)
    SGPA = SGPA + (7* credit);
else if (mark >= 50 && mark <= 59)
    SGPA = SGPA + (6* credit);
else if (mark >= 40 && mark <= 49)
    SGPA = SGPA + (5* credit)
else
  System.out.println("Failed in subject" +(j+1));
}
void Display()
{
System.out.println("Details of student");
System.out.println("Name :" +name);
System.out.println("USN :" +USN);
System.out.println("SGPA of student" + (SGPA/Credits));
}
}
```

```
public class sgpa
{
  public static void main (String args[])
  {
    Student s1 = new Student();
    s1.Details();
    s1.Display();
  }
}
```

* Output :-

```
Enter the USN the student
1BM19CS138
Enter the Name of student
praveen
Enter no of subjects
2
Enter the details of the subjects :
Enter credits allotted to subject 1
4
Enter marks in the subject 1
78
Enter credits alloted to subject 2
4
Enter marks in subject 2
87
Details of student
Name : praveen
USN : 1BM19CS138
SGPA of student 8.5
```

16/10/20

LAB-3

SAI PRAVEEN MARNI
1BM19CS138

Create a class Book which contains four members : name, author, price, num_pages. Include a constructor to set the values for the members. Include a tostring() method that could display the complete details of book. Develope a JAVA Program to create n book objects.

```
import java.util.*;
class Book {
    String name;
    String author;
    double price;
    int num_pages;
Scanner in = new Scanner(System.in);
book(){
  System.out.println("Enter name of the book");
     name = in.nextLine();
  System.out.println("Enter name of the author");
     author = in.nextLine();
  System.out.println("Enter price of the book");
     price = in.nextDouble();
  System.out.println("Enter no. of pages in book");
     Num_pages = in.nextInt();
}
  Public String toString() {
return("Book name : "+name+" Author : "+author+" Price : "+
                price+"Number of pages : "+Num_pages);
}
}
class bookdetail{
  Public static void main(String[] args){
     int i, n;
  Scanner in = new Scanner(System.in);
```

```
System.out.println("Enter no of objects to be created");
    n = int.nextInt();
    book obj[];
    obj = new book[n];
  for(i=0;i<n;i++){
      obj[i] = new Book();
  }
  System.out.println("The details of the books are:");
      for(i=0;i<n;i++){
   System.out.println(obj[i].toString());
   }
  }
}
```

Output :

```
Enter the no of objects to be created:
 2
Enter name of the book:
 wolf hall
Enter name of the author:
 Hillary mantel
Enter price of the book:
   650
Enter no. of pages in the book:
   1500
Enter name of the book:
 The corrections
Enter name of the author:
 Jonathan Franzen
Enter price of the book:
   450
Enter no. of pages in book:
   899
```

The details of the books are :

Book name : wolf hall Author : Hallary mantel Price : 650 No of pages = 1500

Book name : The corrections Author : Jnathan Franzen Price : 450 No of pages = 577

6/11/20

LAB-4

SAIPRAVEEN MARNI
1BM19CS138

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 class extends the class shape. Each one of classes contains only the method printArea() that prints area of given shape.

```
import java.util.*;
abstract class shape
{
  int a=3, b=4;
  abstract public void print_area();
}
class rectangle extends shape
{
    public int area_rect;
    public void print_area()
    {
      area_rect = a*b;
System.out.println("The area of rectangle is " + area_rect);
    }
}
  class triangle extends shape
{
      int area_tri;
    public void print_area()
{
```

```
area_circle = (int)(3.14 * a * a);
System.out.println("The area of circle is :" + area_circle);
}
}
class circle extends shape
{
int area_circle;
Public void print_area()
{
area_circle = (int)(3.14 * a * a);
System.out.println("The area of circle is :" + area_circle);
}
}
public class rtc
{
public static void main (String args[])
{
rectangle r = new rectangle();
r.print_area();
triangle t = new triangle();
t.print_area();
circle r1 = new circle();
r1.print_area();
}
}
```

Output :

The area of rectangle is : 12

The area of triangle is : 6

The area of circle is : 28

06/11/20

LAB - 5

SAIPRAVEEN MARNI
1BM19CS138

Develop a Java program to create a class Bank that maintains two kinds of accounts for its customers, one called savings account and 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 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 acheive following tasks:

- Accept deposit from customer and update the balance.
- Compute & deposit interest. Display the balance.
- Permit withdrawal and update the balance. Check for minimum balance, impose penalty if necessary and update the balance.

```
import java.util.Scanner;
class Account
{
String name;
int accountNo;
String accountType;
double balance;
Account (String name, int accountNo, String accountType, double
                                                    balance) {
this.name = name;
this.accountNo = accountNo;
this.accountType = accountType;
```

```
    }
    Void DisplayStatus()
{
System.out.println("**"+this.accountType+"**");
System.out.println("Name:"+this.name);
System.out.println("Account No:"+this.accountNo);
System.out.println("Account Type:"+this.accountType);
System.out.println("Balance:", +this.balance);
}
}
    class SavAcct extends Account
{
    double depositAmount;
    double withdrawamount;
SavAcct(String name, int accountNo, String accountType,
                                        double balance)
{
 super(name, accountNo, accountType, balance);
}
Static Scanner input = new Scanner(System.in);
 Private void checkBalance()
{
   if(balance<0)
 {
System.out.println("Transcation is not possible. Balance less
                                        than zero");
   balance += withdrawamount;
   withdrawamount = 0;
   withdraw();
 }
}
   void CalInterest()
 {
```

```
System.out.println("Interest to be added");
System.out.println("Amual rate of interest : 4.1%");
System.out.println("Enter tenure in terms of year");
    int tenure = input.nextInt();
balance = balance * math.pow(1.04, tenure);
}
void Deposit()
{
System.out.println("Enter the Deposit Amount");
depositAmount = input.nextDouble();
    balance += depositAmount;
}
void Withdraw()
{
System.out.println("Enter the withdrawal amount");
withdrawmount = input.nextDouble();
    balance -= withdrawmount;
    checkBalance();
System.out.println("withdrawal Amount=" + withdrawmount);
}
}
class CurrAcct extends Account
{
    double minBalance = 1000;
    double depositAmount;
    double withdrawmount;
static Scanner input = new Scanner(System.in);
CurrAcct(String name, int AccountNo, String accountType, double
                                                    balance)
{
    Super(name, accountNo, accountType, balance)
}
```

```
private void checkBalance()
{
    if (balance < minbalance)
{
System.out.println("Transcation not possible. Balance is
                    minimum balance,");
    balance += withdrawamount;
System.out.println("Do u still want to be transcation with
                    added charges");
    String ans = input.next();
    if (ans.toLowerCase().equals("yes"))
    {
    balance -= (withdrawamount + (0.05 * withdrawamount) + 1000)
System.out.println("ALERT: Negative balance. In" Service charg
                    added :" + (0.05 * withdrawamount));
}
  else {
        withdrawamount = 0;
  }
}
}
void Deposit()
{
  System.out.println("Enter Deposit amount");
    depositAmount = input.nextDouble();
      balance += depositAmount;
  }
  void Withdraw()
{
  System.out.println("Enter the Withdrawal Amount");
    withdrawamount = input.nextDouble();
      balance -= Withdrawamount;
      checkBalance();
} System.out.println("withdraw Amount =" + Withdrawamount);
}
```

```
public class BankTest
{ public static void main (String args[])
  {
    Scanner in = new Scanner (System.in);
    System.out.println("Enter the name");
    String name = in.next();
    System.out.println("Enter AccountNo");
    int num = in.nextInt();
  int i=0;
while (i<2)
{
System.out.println("Enter the account type \n curr-current acc.
                    \n sav-savings acct. \t And Balance.");
  String type = in.next();
  if(type.equals("curr"))
  {
  double bal = in.nextInt();
  CurrAcct cl = new CurrAcct(name, num, "Current Account, bal);
    cl.DisplayStatus();
    cl.Deposit();
    cl.Display status();
    cl.withdraw();
    cl.Displaystatus();
}
  else if (type.toLowerCase().equals("sav"));
  {
  double bal = in.nextInt()
  SavAcct sl = new SavAcct (name, num, "Saving Account", bal);
    sl.DisplayStatus();
    sl.Deposit();
    sl.Display Status();
```

```
            s1.calInterest();
            s1.DisplayStatus();
        }
        i++;
    }
    in.close();
}
}
```

Output :

```
Enter the name
praveen
Enter the account no.
1234567890
Enter the accountType
Curr - current acc.
Sav - Savings acct.        And Balance.
curr
560000
** Current Account **
Name : praveen
Account no. : 1234567890
AccountType : Current Account
Balance : 560000.0
Enter the Deposit Amount
12000
** Current Account **
Name : praveen
Account No : 1234567890
Account Type : Current Account
Balance : 57200000
Enter the withdrawal Amount
34000
withdraw amount = 34000.0
```

** Current Account **

Name : praveen

Account No : 1234567890

Account Type : Current Account

Balance : 538000.0

Enter the account type

curr - Current acc.

Sav - Savings acct. And Balance.

sav

12345

** Savings Account **

Name : praveen

Account No : 1234567890

Account Type : Saving Account

Balance : 12345.0

Enter the Deposit Amount

123

** Savings Account **

Name : praveen

Account No : 1234567890

Account Type : Savings Account

Balance : 12468.0

Enter the withdrawal Amount

3456

Withdraw Amount = 3456

** Savings Account **

Name : praveen

Saccount No : 1234567890

Account Type : Savings Account

Balance : 9012.0

Interest To Be Added

Annual rate of Interest : 4%

Enter tenure in terms of years : 3

** Savings Account **

Name : praveen

Account No : 1234567890

Account Type : Saving Account

Balance : 10137.274368