18/12/25

Quadratic Equation

```
import java.util.*;
class Quadratic
{
public static void main(String args[])
{
  double a;
  double b;
  double c;
  d=Math.pow(b,2)-(4*a*c);
  e=Math.pow(Math.pow(b,2)-(4*a*c),2);
  Scanner S1=new Scanner(System.in);
System.out.println("enter the coefficients");
a=S1.nextDouble();
b=S1.nextDOuble();
c=S1.nextDouble();
if(a==0)
{
 System.out.println("Enter a valid equation");
}
else{
    double d
    double e;
if(d>=0){
 System.out.println("the roots are real
  and equal");
System.out.println("The roots are"+
  (-1*b/(2*a))+" and "+(-b/(2*a));
}
else if
{
```

```
System.out.println("The roots are
real and unequal");
System.out.println("The roots are"+
((-1*b+e)/(2*a))+and((-1*b-e)/(2
}
else if(d<0)
{
d=d*(-1);
d=Math.pow(d,0.5)
System.out.println("The roots are
real and unequal");
System.out.println("The roots are"+
(-1*b/2*a)+"+i"+(e/(2*a))+"and
System.out.println("The roots are"+
(-1*b/2*a)+"-i"+(e/(2*a)));
}
}
}
```

Output

Enter the coefficients

1.0

1

2

Enter valid eqn

1

1

2

1

The roots are real and equal

-1, 0

Date __/__/__
Page

III The roots are $\frac{1}{3}$, $\frac{1}{2}$

The roots are real and unequal
$-2, 0$.

Re
18/12/2022

111124

LAB-2

1. 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 method to calculate SGPA of a student.

```
import java.util.*;
class Student
{
String usn;
String name;
int [] credits;
int A[] marks;
public void Acceptdetails(int n)
{
Scanner s1 = new Scanner(System.in);
System.out.println("enter USN");
Usn = s1.next();
System.out.println("enter name");
Name = S1.next();
credits = new int [n];
marks = new int [n];
for(int i=1; i<= credits.length; i++
{
System.out.println("enter credits for
  Subject"+ i+":");
Credits[i-1] = s1.nextInt();
System.out.println("enter marks for sub
i+":");
marks[i-1] = s1.nextInt();
}
}
```

```
public void DisplayDetails()
{
System.out.println("usn");
System.out.println(name);
for(int i=1; i<=marks.length; i++)
{
System.out.println("Subject "+i+" marks");
}
}
public float SGPA()
{
int totalCredits=0;
int totalGradePoints=0;
for(int i=1; i<=marks.length; i++)
{
totalCredits+=credits[i-1];
int temp=gradePoint(marks[i-1]);
gradePoints+=Credits[i-1]*temp;
}
return gradePoints/totalCredits;
}
public int gradePoint(int marks)
{
if(marks>=90)
return 10;
else if(marks>=80)
return 9;
else if(marks>=70)
return 8;
else if(marks>=60)
return 7;
else if(marks>=50)
return 6;
```

```
else if (marks >= 40)
    return 5;
else
    return 0;
}
}

class Student detail
{
public static void main (String args[])
{
Student s = new Student();
s.Accept Detail();
s.Display Detail();
System.out.println("SGPA" + s.SGPA);
```

Output

~~Enter~~ enter usn :
1BM22CS154
Enter Credits for subject 1:
4
enter marks for subject 1:
90
enter credits for subject 2:
4
enter ~~Credit~~ marks for subject 2:
70
enter Credits for subject 3:
3
enter marks for subject 3:
90
1BM22CS154
Subject 1 - 90
Subject 2 - 70

Date __/__/__
Page 1

Subject: 390
Scripts & Q

22/11/24

8/11/24

Lab-3

Create a class Book which contains four members: name, author, price, num_pages. Include constructor to set the values for the members. Include methods to set and get the details of the objects. Include a to-string method that could display the details of the book. Develop a java program to create n book objects

```
import java.util.Scanner;
class BookDemo
{
 String name;
 String author;
 double price;
 int num_pages;
 Scanner sc = new Scanner(System.in);
BookDemo(){ }
BookDemo(String name, String author,
double price, int num_pages)
{
 this.name = name;
 this.author = author;
 this.price = price;
 this.num_pages = num_pages;
}
void AcceptDetails()
{
 System.out.println("enter name");
```

```
name=sl.next();
System.out.println("enter author");
author=sl.next();
System.out.println("enter price");
price=sl.nextInt();
System.out.println("enter number of pages");
num_pages=sl.nextInt();
}
public String toString()
{
return("Book Details:\nname:"+name
+"\nauthor:"+author+"\nprice:"+
price+"\nnumber of pages:"+num_pg
}
}
class Book
{
public static void main(String args[])
{
Scanner sl=new Scanner(System.in);
int n=System.out.println("enter number
of books");
int n=sl.nextInt();
BookDemo b[]=new BookDemo[n];
for(int i=0;i<n;i++)
{
b[i]=new BookDemo();
b[i].AcceptDetails();
}
System.out.println("Details of Books:");
for(int i=0;i<n;i++)
{
```

```
System.out.println(b[i]);
}
}
}
```

Output: enter number of books: 2
enter name
abc
enter author
xyz
enter price
90
enter number of pages
100
enter name
rst
enter author
uvw
enter price
80
enter number of pages
200
Book Detail
name: abc
author: xyz
price: 90
Number of pages: 100
name: rst
author: uvw
price: 80
number of pages: 200

22/1/24

8/11/24

Date __/__/__
Page _____

Lab-3

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

```
import java.util.Scanner;
abstract class Shape
{
public abstract void printArea();
}
class Rectangle extends Shape
{
private int length;
private int breadth;
public Rectangle(int length, int breadth
{
this.length = length;
this.breadth = breadth;
}
public void printArea()
int area = length * breadth;
System.out.println("area of rectangle:"+are
}
}
```

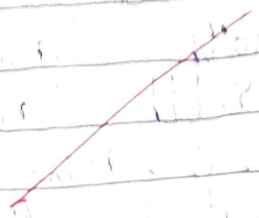
```
class Triangle extends Shape
{
  private int base;
  private int height;
public Triangle(int base, int height)
{
  this.base = base;
  this.height = height;
}
public void printArea()
{
double area = 0.5 * base * height;
System.out.println("the area of Triangle
  " + area);
}
}
class Circle extends Shape
{
  private int radius;
public Circle(int radius)
{
  this.radius = radius;
}
public void printArea()
{
double area = Math.PI * radius * radius;
System.out.println("Area of circle" + area);
}
}
class Main
{
```

```
public Static void main (String args[])
{
  Scanner S1 = new Scanner (System.in);
System.out.println("enter length and
breadth of the rectangle");
int length = S1.nextInt();
int breadth = S1.nextInt();
Rectangle r1 = new Rectangle (length, breadth
System.outs.println("enter the base and
height of the ~~rectangle~~ triangle");
int base = S1.nextInt();
int height = S1.nextInt();
Triangle t1 = new Triangle (base, height);
System.out.println("enter the radius of
the circle");
int radius = S1.nextInt();
Circle C1 = new Circle (radius);
r1.printArea();
t1.printArea();
C1.printArea();
```

Output

enter length and breadth of the rectangle
10
15
enter the base and height of the ~~rectangle~~ triangle
20
30
enter the radius of the circle
30

area of rectangle: 150.
area of triangle: 300.0
area of circle: 2827.4333882308138



22/11/24.

Date __/__/__
Page

Lab-4

Write a program that demonstrates handling of exceptions in inheritance tree. Create a base class "Father" and a derived class Son which extends the base class. In father class, implement a constructor which takes the age and throws the ~~WO~~ WrongAge() when the input age < 0. In Son class, implement a constructor that takes both father and son's age and throws an exception if Son's age >= father's age

```
import java.util.Scanner;
class WrongAge extends Exception
{
    WrongAge(String message)
    {
        super(message);
    }
}
class Father
{
    private int age;
    Father(int age) throws WrongAge
    {
        if(age<0)
        {
            throw new WrongAge("Age cannot be negative");
        }
```

```
    this.age = age;
  }
  int getAge()
  {
    return age;
  }
}
class Son extends Father
{
  private int SonAge;
  Son(int fatherAge, int SonAge)
  throws WrongAge
  {
    super(fatherAge);
    if (SonAge >= fatherAge)
    {
      throw new WrongAge("Son's age
      should be less than father's age");
    }
    this.ageSonAge = SonAge;
  }
int getSonAge()
{
  return SonAge;
}
}
class SonFather
{
  public static void main(String args[]
  Scanner s1 = new Scanner(System.in);
  try
  {
```

```
System.out.println("enter father's age");
int fatherAge = sc.nextInt();
System.out.println("enter son's age");
int sonAge = sc.nextInt();
Son son = new Son(fatherAge, sonAge);
System.out.println("Father's age:" +
    son.getAge());
System.out.println("Son's age:" +
    son.getSonAge());
} catch (WrongAge e) {
    System.out.println("Exception" +
        e.getMessage());
} finally {
    sc.close();
}
}
```

Output

1) enter father's age
45
enter son's age
12
father's age: 45
Son's age: 12

2) enter father's age
-40
enter son's age
-20
Exception: Age cannot be negative

3)
enter father's age
12
enter Son's age
30
Exception: Son's age should be less than father's age

LAB-4

Creat a package CIE which has 2 classes Student and Internals. The class Student has members like USN, Name, Sem. The class Internals has an array of marks. Creat another package SEE which has the class External which is a derived class of Student. This class has an array of marks scored in SEE. Import both the 2 packages in a file that shows the SEE marks scored and

```
package CIE;
public class Student
{
public String USN, Name;
public int Sem;
public Student (String USN, String
Name, int Sem) {
{
this.USN = USN;
this.Name = Name;
this.Sem = Sem;
}
}
```

```
package CIE;
public class Internals extends Student
{
  public int marks[] = new int[5];
  public Internals (String USN, String
name, int sem, int[] m){
```

```
}
super (USN, Name, Sem)
this.marks=m
}
}
Package SEE;
import CIE.Student;
Public class External extends CIE.Student
{
  public int fy[] = new int [5];
  public External(String USN, String name,
    int Sem, int[] fy);
  {
    super(Usn, nam, Sem);
    this.fy = fy
  }
}
import SEE.java.util.*
import CIE.Student;
import CIE.External;
import SEE;
public class main
{
  public static void main (String args[])
  int fm=0;
  Scanner s1 = new Scanner (System.in);
  System.out.println("Enter the no. of Student");
  int n = s1.nextInt();
  Internal[] in = new Internals[n];
  External[] ems = new External[n];
  Student.[] stu = new Student[n];
  for (int i=0; i<n; i++)
```

```
System.out.println("Enter details of
 Student "+(i+1)+" ");
System.out.println("enter name");
name = sl.next();
System.out.println("Enter USN");
String USN = sl.next();
System.out.println("Enter Semester");
int sem = sl.nextInt();
int marks[] = new int[5];
System.out.println("Enter marks");
for(int j=0; j<5; j++)
{
  System.out.println("Enter marks"
        +(j+1));
  marks[j] = sl.nextInt();
}
  System.out.println("Enter external marks");
for(int j=0; j<5; j++)
{
  System.out.println("Enter external
    marks of "+(j+1));
    marks[j] = sl.nextInt();
}
stu[i] = new Student(usn, name, sem);
im[i] = new Internal(usn, name, sem, imarks);
em[i] = new External(usn, name, sem, imarks);
}
System.out.println("Final marks");
for(int i=0; i<n; i++)
{
  System.out.println("Student "+(i+1));
  System.out.println("Name "+stu[i].name);
```

```
System.out.println("USN"+Stu[i].USN);
System.out.println("Sem"+Stu[i].sem);
for(int j=0;j<5;j++)
{
 fm+=Im[i].m[j]+em[i].sm[j];
 System.out.println("Final marks of
   course"+(j+1)+":"+fm);
fm=0;
}
}
}
}
```

Output

Enter the no. of student
ABC
java.util.mismatch Exception

Enter the no. of student
1
Enter the detail
Enter name
XYZ
Enter USN
1BM16
Enter Sem
4
Enter the internal marks
Enter the internal marks of 1: 40
Enter the internal marks of 2: 45
Enter the internal marks of 3: 50
Enter the internal marks of 4: 44

Enter the internal marks of 5:50

Enter the external mark of 1:45
Enter the external marks of 2:50
Enter the external marks of 3:46
Enter the external marks of 4:30
Enter the external marks of 5:43

Final marks obtained

Student 1:
Name: XYZ
USN: 1BMbb
Sem: 4
Final mark of course 1: 85
Final marks of course 2: 95
Final marks of course 3: 76
Final marks of course 4: 74
Final marks of course 5: 93

late submission

5/2/2024

Date / /
Page

LAB-5

Write a program which creates two threads, one thread displaying "BMS College of Engineering" once every ten seconds and another displaying "CSE" once every two seconds.

```
class BMSThread extends Thread
{
  public void run()
  {
    while(true)
    {
      System.out.println("BMS college of
        Engineering");
      try
      {
        Thread.sleep(10000);
      }
      catch (InterruptedException e)
      {
        e.printStackTrace();
      }
    }
  }
}
class CSEThread extends Thread
{
  public void run()
  {
    while (true)
```

```
System.out.println("CSE");
try
{
    Thread.sleep(2000);
}
catch (InterruptedException e)
{
    e.printStackTrace();
}
}
}
}
```

Output

BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
.
.

```
class BMSThreadDemo
{
    public static void main(String[] args)
    {
        BMSThread bmsThread = new BMSThread
```

```
CSEThread cseThread = new CSEThread();
bmsThread.start();
cseThread.start();
}}
```

Output

Bms college of Engineering
CSE
CSE
CSE
CSE
CSE
Bms college of Engineering
CSE
CSE
CSE
CSE
CSE.

Date __/__/__
Page _____

11/2/24

Lab-6

Bank Program

```
import java.util.Scanner;
Class CanaraBank
{
  String customername;
  String accountnumber;
  String type;
  double balance;
  CanaraBank(String customername, String
             Accountnumber, String type);
  {
    this.customername = customername;
    this.accountnumber = accountnumber;
    this.type = type;
  }
  Void minimumBalance()
  {
    if (this.balance<1000)
    {
      System.out.println("Cannot
      not follow the rules of minimum
      balance, So Rs 20 is deducted");
      balance = balance-20;
    }
  }
}
Class SavingsAccount extends CanaraBank
{
  double money;
  double interest;
```

```
double depositperiod;
SavingsAccount(String customername, String
        acountnumber, string type,
        double moneyin, double
        interest, double depositperiod)
{
  super(customername accountnumber, type);
   this.moneyin = moneyin;
   this.interest = interest;
   this.depositperiod = depositperiod;
}
Void displayBalance()
{
   balance = moneyin *(1+depositperiod^2
                interest);
   System.out.println("The amount in
          Bank after" + depositperiod +
          "is" + balance);
}
Void withdraw(double with)
{
   balance = balance - with;
    this.minimumBalance();
   System.out.println("the amount
        left in Savings account is" + balance);
}
Void deposit()
{
   System.out.println("Enter the amount
                you want to deposit");
   double deposit;
   Scanner S1 = new Scanner(System.in);
```

Date __/__/__
Page _____

```
deposit= S1.nextDouble();
balance= balance+deposit;
this.minimumbalance();
System.out.println("the amount in Bank
                    Savings Account is" + balance
}
}
class CurrentAccount extends CanaraBank
{
  double moneyin;
  CurrentAccount(String Customername,
  String accountnumber, String type,
  double moneyin)
  {
    super(customername, accountnumber,
          type);
    this.moneyin = moneyin;
    balance= moneyin;
    this.minimumBalance();
  }
  void checkfacilities()
  {
    System.out.println("the account
        has check facilities");
  }
  void depositthroughCheck(int money);
  {
    balance= balance+money;
    this.minimumBalance();
    System.out.println("the new
      Balance is" + balance);
  }
```

```
void displayBalance()
{
  System.out.println("The amount in
                      Bank with "+this.
                      "account after "+
                      this.balance);
}
}
class Bank
{
  public static void main(String[] args)
  {
    SavingsAccount A = new SavingsAccount
    ("Mithun", "1000", "Savings", 900.0);
    A.displayBalance();
    A.withdraw(20.00);
    A.deposit;
    CurrentAccount B = new CurrentAccount
    ("Manish", "1001", "Current", 1700);
    B.checkFacilities();
    B.depositThroughCheque(500);
    B.displayBalance();
  }
}
```

Output:

The amount in Bank after 1.0 is 990.
You do not follow rules of minimum balance So 20 rupees is deducted
The amount left in Savings account is 950.
Enter the amount you want to deposit

Date __/__/__
Page _____

10.
You do not follow rule of minimum balance so 20 rupee is deducted
The amount in bank account is 940
This account has check facilities
The new balance is 20/2200
The amount in Bank Current Account is 2260.0.

19/12/24