

Week 2

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

```
class QuadraticEquation
```

{

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

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

```
        System.out.println("Enter the value of a:");
```

```
        double a = input.nextDouble();
```

```
        System.out.println("Enter the value of b:");
```

```
        double b = input.nextDouble();
```

```
        System.out.println("Enter the value of c:");
```

```
        double c = input.nextDouble();
```

```
        if (a == 0)
```

{

```
            System.out.println("Invalid Input");
```

}

```
        else
```

{

```
            double d = (b * b) - (4 * a * c);
```

```
            if (d > 0)
```

{

~~double r1 = (-b + Math.sqrt(d)) / (2 * a);~~~~double r2 = (-b - Math.sqrt(d)) / (2 * a);~~

```
            System.out.println("The roots are real and " +
```

distinct

```
                + r1 + " and " + r2);
```

}

```
        else if (d == 0)
```

```
{
```

```
    double r1 = (-b) / (2 * a);
```

```
    System.out.println("a" + " " + b + " " + c + " " + d);
```

```
    System.out.println("The roots are real and equal  
    " + r1 + " and " + r1);
```

```
}
```

```
else
```

```
{
```

```
    double r1 = (-b) / (2 * a);
```

```
    double r2 = Math.sqrt(Math.abs(d)) / (2 * a);
```

```
    System.out.println("The roots are imaginary"  
    + " " + r1 + " " + r2 + " " + r1 + " " + r2);
```

```
}
```

```
// else
```

Output

1) Enter the value of a

2

Enter the value of b

3

Enter the value of c

4

The roots are real and distinct

- 2.192235 and - 2.28077

1) Enter the value of a

2

Enter the value of b

4

Enter the value of c 2

The roots are real and equal -1.0 and -1.0

3) Enter the value of a

1

Enter the value of b

2

Enter the value of c

3

The roots are distinct and imaginary $r_1 = -1.0 + i 1.41421356$

$$r_2 = -1.0 - i 1.41421356$$

4) Enter the value of a

0

Enter the value of b

1

Enter the value of c

2

The Invalid input

880

18/11/2022

Exercise 2

Import java.util.Scanner;

class student

void display (String usn, String name)

{

System.out.println ("USN of the student" + usn);

System.out.println ("Name of the student" + name);

}

void calculateGPA (double [] marks, double [] credits,
int number)

{

double gradepoints [] = new double [number];

double sgpa, sum = 0, tnum = 0

for (i=0; i<number; i++)

{

if (marks [i] >= 90)

gradepoints [i] = 10;

else if (marks [i] >= 80)

gradepoints [i] = 9;

else if (marks [i] >= 70)

gradepoints [i] = 8;

else if (marks [i] >= 60)

gradepoints [i] = 7;

else if (marks [i] >= 50)

gradepoints [i] = 6;

else if (marks [i] >= 40)

gradepoints [i] = 4;

else

gradepoints [i] = 0;

}

for (int i = 0; i < number; i++)

d

sum = credits[i] * gradepoints[i];

y

for (int i = 0; i < number; i++)

d

trnum = credits[i];

y

sgpa = sum / trnum;

System.out.println("SGPA is " + sgpa);

p
p

QIP?

class SGPA d

public static void main (String args []) {

Scanner s = new Scanner (System.in);

System.out.println ("Enter name & roll no of student");

String name = s.next();

String rn = s.next();

student s1 = new student();

System.out.println ("Enter the no of courses");

int number = s.nextInt();

double credits [] = new double [number];

double marks [] = new double [number];

for (int i = 0; i < number; i++)

d

System.out.print ("Credit of subject " + (i+1) + ": ");

credits[i] = s.nextDouble();

System.out.print ("Marks of subject " + (i+1) + ": ");

marks[i] = s.nextDouble();

y

s1.display (name, rn);

s1.calculatesgpa (marks, credits, number);

y

Exercise 2 output

Enter the name and usn of student
athena

1bm21cs255

Enter the number of course

5

Credit of subject 1 : 4

Marks of subject 1 : 93

Credit of subject 2 : 3

Marks of subject 2 : 85

Credit of subject 3 : 3

Marks of subject 3 : 79

Credit of subject 4 : 1

Marks of subject 4 : 98

Credit of subject 5 : 1

Marks of subject 5 : 100

USN of the student 1bm21cs255

Name of the student ~~athena~~

SGPA is 9.25

Exercise 3

```
import java.util.*;  
import java.lang.*;
```

```
class Book
```

```
{  
    String name, author; int price, num_pages;
```

```
    void getall()
```

```
{  
    Scanner sc = new Scanner(System.in);
```

```
    System.out.println("Enter book name");
```

```
    name = sc.next();
```

```
    System.out.println("Enter author name");
```

```
    author = sc.next();
```

```
    System.out.println("Enter price");
```

```
    price = sc.nextInt();
```

```
    System.out.println("Enter no of pages");
```

```
    num_pages = sc.nextInt();
```

```
}
```

```
    public String toString()
```

```
{
```

```
    return name + " " + author + " " + price + " " + num_pages;
```

```
"";
```

```
}
```

```
    void display(Book o)
```

```
{
```

```
    System.out.println(o);
```

```
}
```

```
}
```

class BookVck

2

public static void main (String args [])

2

Scanner in = new Scanner (System . in);

System . out . println (" Enter the no. of book objects ");

int n = in . nextInt ();

Book [] ob = new Book [n];

for (int i = 0; i < n; i ++)

ob [i] = new Book ();

for (int i = 0; i < n; i ++)

2

ob [i] . getval (); } }

for (int i = 0; i < n; i ++)

2

ob [i] . display (ob [i]);

2

O/P

CS
1/12/20 ✓

Exercise 3 output

Enter the no of book objects

2

Enter book name

abc

Enter author name

tom

Enter price

100

Enter No of pages

600

Enter book name

xyz

Enter author name

sam

Enter price

500

Enter No of pages

400

abc tom 100 600

xyz sam 500 400

SS19
1/12/2021

Week 4

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.

CODE

abstract class shape {
 double a, b;

shape (int x, int y)

{

a = x;

b = y;

}

abstract void printArea();

}

→ Class rectangle extends shape {
 rectangle (int x, int y)
 {
 super (x, y);
 }

void printArea()

d

System.out.println("Area of rectangle is " + (a+b));

)

}

→ class triangle extends shape1

triangle (int x, int y)

{

super(x,y);

)

void printArea()

d

System.out.println("Area of triangle is " + (0.5*a*b));

)

}

→ class circle extends shape1

circle (Pnt x, int y)

d

super(x,y);

)

void printArea()

d

System.out.println("Area of circle is " + (3.14*a*a));

)

)

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

{
rectangle r1 = new rectangle (10, 20);
triangle t1 = new triangle (5, 10);
circle c1 = new circle (3, 0);

shape r1;

r1 = r1;

r1.printArea();

r1 = t1;

r1.printArea();

r1 = c1;

r1.printArea();

}
}

OUTPUT

Area of rectangle is 200.0

Area of triangle is 25.0

Area of circle is 28.259999

~~SO
8/12/2022~~

Week 5

```
import java.util.*;  
import java.lang.*;
```

```
class Account
```

L

```
public String acname;  
public double acno;  
public int actype;  
public double balance;  
public void getdata (String name, double no, int  
type, double bal)
```

L

```
acname = name;
```

```
acno = no;
```

```
actype = type;
```

```
balance = bal;
```

 }

```
class Savings extends Account
```

L

```
public void deposit (double amt)
```

L

```
balance = balance + amt;
```

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

Y

public void withdraw (double amt)

d : balance + anti

balance = balance +
systems out problem (balance).

3

public void interest (int time, int no)

三

```
double intn = balance * (1 + b / no);
```

ints = Math. pow (int₁, (time + no));
" " tangent call

System.out.println ("Interest calculated
+ int);

balance = balance + int*i*;

```
System.out.println("The new balance is " + balance);
```

Y

class current extends Account

d

public void deposit (double amt)

d

balance = balance + amt;

System.out.println(balance);

30

~~public void withdraw (double amt)~~

d

~~balance~~ = balance - amt.

System-out printer (balanci).

chart (balance);

3

```
- public void check(double amt)
  {
    if (amt < 10000)
      balance = balance - 500;
  }
```

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

}

}

}

```
class Main
```

```
public static void main (String args[])
  {
    Scanner sc = new Scanner (System.in);
    int temp = 1;
```

```
    while (temp == 1)
```

```
    {
      double amt = 0;
```

```
      System.out.println ("Enter name");
      sc.next();
```

```
      String name = sc.nextLine();
```

```
      System.out.println ("Enter accno");
```

```
      double no = sc.nextDouble();
```

```
      System.out.println ("Enter acctype\n 0 for Saving\n 1 for Current");
    }
```

~~int type = sc.nextInt()~~

do

h

System.out.println ("Enter balance");
amt = sc.nextDouble();

y

while (type == 1 && amt < 10000);
if (type == 0)

d

savings = new Savings();

s.getdata (name, no, type, amt);

System.out.println ("In 1. Deposit In 2. Withdraw
In 3. Interest");

end temp3 = sc.nextInt();

if (temp3 == 1)

d

System.out.println ("Enter Amount");

double amt1 = sc.nextDouble();

s.deposit (amt1);

f

else if (temp3 == 2)

h

System.out.println ("Enter Amount");

double amt1 = sc.nextDouble();

s.withdraw (amt1);

f

else if (temp3 == 3)

d

System.out.println ("Enter time period");

int tb = sc.nextInt();

System.out.println ("Enter no. of times");

int nob = sc.nextInt();

s. Interest (tb, nob))

}

else if (type == 1)

h current c = new current ()

c. getdata (new name, no, type, amt);

System.out.println ("1. Deposit 2. Withdraw");

int temp3 = sc.nextInt();

if (temp3 == 2)

d

System.out.println ("Enter Amount");

double amt1 = sc.nextDouble();

c. withdraw (amt1);

}

System.out.println ("To continue 1 else 0");

temp = sc.nextInt();

Y

Y

Y

output

Enter name

Karen

Enter account no.

100

Enter activity

0 for Savings

1 for Current

0

Enter balance

1000 0

1. Deposit

2. Withdraw

3. Interest

1

Enter Amount

20000

30000. 0

✓ \$50
29/12/2021

Week 7

PROGRAM

Write a program that demonstrates handling of exceptions in inheritance tree. Create a base class called "Father" and derived class called "Son" which extends the base class. In Father class, implement a constructor which takes the age and throws the exception 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 is \geq father's age.

CODE

```
import java.util.*;  
class Wrongage extends Exception  
{  
    int detail;  
    Wrongage (int d)  
    {  
        detail = d;  
    }  
    public String toString()  
    {  
        return "Entered wrong age is [" + detail + "]";  
    }  
}
```

```
→ class Father {
    int f;
    Scanner in = new Scanner(System.in);
    Father() {
        {
            System.out.println("Enter father age");
            f = in.nextInt();
        }
    }
    void checkage () throws Wrongage {
        if (f < 0)
            throw new Wrongage(f);
    }
    System.out.println("Father age positive");
}
```

```
→ class Son extends Father {
    int s;
    Scanner in = new Scanner(System.in);
    Son() {
        super();
        System.out.println("Enter son age");
        s = in.nextInt();
    }
}
```

void checkage () throws Wrongage
d

super.checkage();

if (s < 0)

d

throw new Wrongage(f);

}

System.out.println("Son age positive");

}

void checkage () throws Wrongage

d

if (s > b)

d

throw new Wrongage(s);

}

System.out.println("Father-Son age correct");

l

l

class Newdemo

public static void main (String args[])

d

int b,s;

Father fath = new Father();

Father s;

s = fath;

try 2

m.checkage();

}

catch (Wrongage e) {

System.out.println (" Father's age is wrong "+e);

}

Son sn = new Son();

m=sn;

try 3

Sn.checkages();

m.checkage();

}

catch (Wrongage e) {

System.out.println (" Son's age is wrong "+e);

}

}

}

OUTPUT

Enter father age

-10

Father age entered wrong age is [-10] // Father's age is negative

↳ Enter Father age 40

↳ Enter Son age 50

Son age entered wrong age [50] // Son is older than father

50
50
50

PROGRAM

10

package CIE;

```
public class Student {  
    public String usn;  
    public String name;  
    public int sem;  
}
```

Class Internals extends Student &

```
public int [] marks = new int [5];  
}
```

package SEE;

import CIE.Student;

public class External extends Student &

```
public int [] marks = new int [5];  
}
```

Import java.util.Scanner;

import CIE.Internals;

import SEE.External;

public class Main &

public static void main (String [] args) &

Scanner <= new Scanner (System.in);
int n, i, j;

System.out.println ("Enter the number of students");
n = sc.nextInt();

Internals [] enter = new Internals [n];
Externals [] enter = new Externals [n];

for (i=0; i<n; i++) {

Inter [i] = new Internal ();

enter [i] = new External ();

System.out.println ("Enter the details of student " + (i+1));

System.out.println ("Enter USN : ");

Inter [i].usn = sc.nextInt();

enter [i].usn = Inter [i].usn;

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

Inter [i].name = sc.next();

enter [i].name = Inter [i].name;

System.out.println ("Enter USN : semester ");

Inter [i].sem = sc.nextInt();

enter [i].sem = Inter [i].sem;

System.out.println ("Enter the internal marks of 5 courses having 3 credits ");

for ($j=0$; $j < 5$; $j++$) {

 print(i).marks[j] = sc.nextInt();

}

System.out.println ("Enter the external marks of 5 courses having 3 credits");

for ($j=0$; $j < 5$; $j++$) {

 enter(i).marks[j] = sc.nextInt();

}

S-O.P.tn ("Details of students with their final marks")

for ($i=0$; $i < n$; $i++$) {

 System.out.println ("Student" + ($i+1$) + ":" +);

 System.out.println ("USN:" + print(i).usn);

 System.out.println ("Name:" + print(i).name);

 System.out.println ("Semester:" + print(i).sem);

 System.out.println ("Final Marks:"),

 for ($j=0$; $j < 5$; $j++$) {

 System.out.print (($j+1$) + ":" + (print(i).marks + enter(i).marks[j] / 2)));

}

}

}

PROGRAM //

class A {

int n;

boolean valueSet = false;

synchronized int get() {

while (!valueSet)

try { wait(); }

catch (InterruptedException e)

}

System.out.println ("Interrupted Exception caught");

this.n = n;

valueSet = true;

System.out.println ("Put: " + n);

notify(); return n;

}

synchronized void put (int n) {

while (valueSet)

try { wait(); }

catch (InterruptedException e)

}

System.out.println ("Interrupted Exception caught");

this.n = n;

valueSet = true;

System.out.println ("Put = " + n);

notify();

}

class Producer implements Runnable {

 Queue q;

 Producer(Q q) {

 this.q = q;

 new Thread(this, "Producer").start();

}

 public void run() {

 int i = 0;

 while (true) { q.put(i++); }

}

}

 Q q;

 Consumer(Q q) {

 this.q = q;

 new Thread(this, "Consumer").start();

}

 public void run() {

 while (true) { q.get(); }

}

}

class JTC {

 public static void main (String args[]) {

 Q q = new Q();

 new Producer(q);

 new Consumer(q);

 System.out.println ("Press Ctrl C to stop");

}

PROGRAM 8

```
class Thread1 extends Thread {  
    public void run() {  
        while (true) {  
            System.out.println("BMS College of Engg.");  
            try {  
                Thread.sleep(10000);  
            } catch (InterruptedException e) {  
                continue;  
            }  
        }  
    }  
}
```

```
class Thread2 extends Thread {  
    public void run() {  
        while (true) {  
            System.out.println("CSE");  
            try {  
                Thread.sleep(2000);  
            } catch (InterruptedException e) {  
                continue;  
            }  
        }  
    }  
}
```

}
}

public class Thread - prog 8 d

```
public static void main (String args [] ) {  
    System.out.println ("Control-C to end");  
    new Thread1().start ();  
    new Thread2().start ();
```

p p

Output

CSE

CSE

CSE

CSE

BMS College of Engineering

CSE