

Lab 1

1. Develop a Java program that prints all real solutions to the quadratic equation $ax^2+bx+c = 0$. Read in a , b , c and use the quadratic formula. If the discriminant b^2-4ac is negative, display a message stating that there are no real solutions.

```
import java.util.Scanner;
import java.lang.Math;
```

```
class solution
{
    public static void main(String args[])
    {
        Scanner S = new Scanner(System.in);
        double a,b,c,r1,r2;
        System.out.println("Enter the values of a,b,c:");
        a = S.nextDouble();
        b = S.nextDouble();
        c = S.nextDouble();
        if (a != 0){
            double d = b*b - 4*a*c;

            if(d > 0){
                System.out.println("Roots are real and distinct");
                r1 = (-b + Math.sqrt(d))/(2*a);
                r2 = (-b - Math.sqrt(d))/(2*a);
                System.out.println("r1 : " + r1 + "r2 : " + r2);
            }
            else if(d == 0){
                System.out.println("Roots are real and equal");
                r1 = (-b)/(2*a);
                System.out.println("roots are : "+ r1);
            }
            else{
                System.out.println("Roots are imaginary");
                r1 = -b/(2*a);
                r2 = Math.sqrt(Math.abs(d))/(2*a);
                System.out.println("Roots are Imaginary");
                System.out.println("r1 : " + r1 + "i" + r2 + "r2 : " + r1 + "-i" + r2);
            }
        }
        else{
            System.out.println("Invalid Input");
        }
    }
}
```

Observation Book

↳ Develop a Java program to solve a Quadratic equation accepting the input gesture from the user and display message for invalid input for imaginary roots.

```
import java.util.Scanner;
import java.lang.Math;

class Solution {
    public static void main (String args[])
    {
        Scanner S = new Scanner (System.in);
        double a, b, c, r1, r2;
        System.out.println ("Enter the values of a, b, c : ");
        a = S.nextDouble();
        b = S.nextDouble();
        c = S.nextDouble();
        if (a != 0)
        {
            double d = b * b - 4 * a * c;
            if (d > 0)
            {
                System.out.println ("Roots are Real and Distinct");
                r1 = (-b + Math.sqrt(d)) / (2 * a);
                r2 = (-b - Math.sqrt(d)) / (2 * a);
                System.out.println ("r1 : " + r1 + " r2 : " + r2);
            }
            else if (d == 0)
            {
                System.out.println ("Roots are real and Equal");
                r1 = (-b) / (2 * a);
                System.out.println ("Roots are : " + r1);
            }
        }
    }
}
```

```

else {
    roots are real and equal
    System.out.println("Roots are Real");
}
}
else {
    System.out.println("Invalid Input");
}
}
}

```

Output:

Enter the values of a,b,c:

1 2 1

Roots are real and equal

roots are : -1.0

Enter the values of a,b,c:

1 5 2

Roots are real and distinct

roots are $r_1 = -0.138$, $r_2 = -0.562$

Enter the values of a,b,c:

0 1 3

Invalid Input

Enter the values of a,b,c:

1 2 3

Roots are Imaginary.

Output images



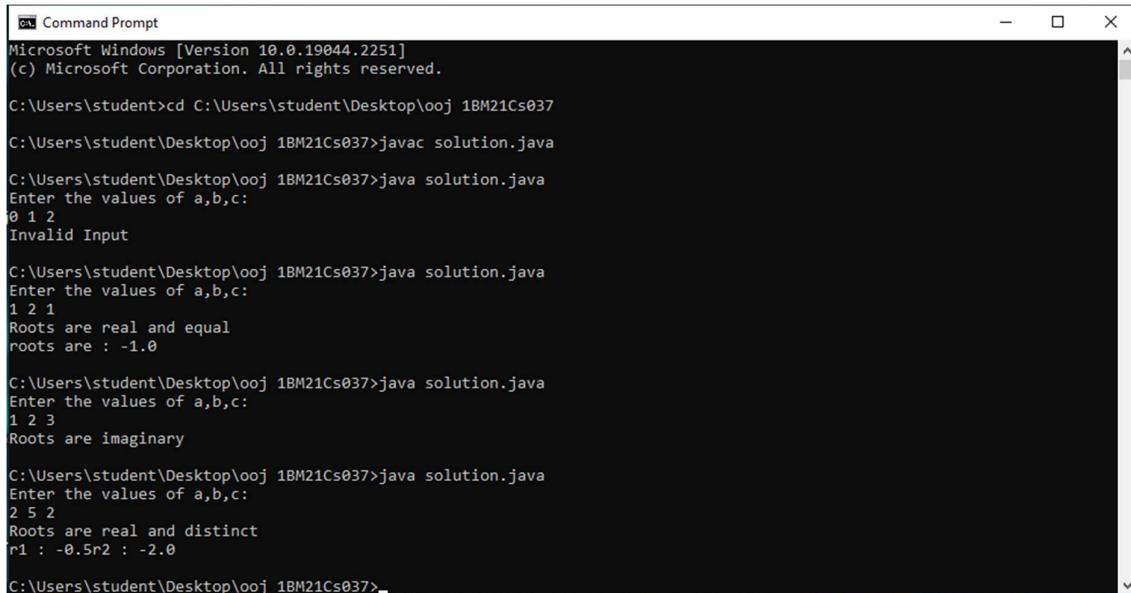
```
Microsoft Windows [Version 10.0.19045.2251]
(c) Microsoft Corporation. All rights reserved.

C:\Users\bmsce>cd C:\Users\bmsce\Desktop\1BM21CS047\week-1

C:\Users\bmsce\Desktop\1BM21CS047\week-1>javac qe.java

C:\Users\bmsce\Desktop\1BM21CS047\week-1>java qe
Enter values of a,b and c
*1 2 3
Roots are imaginary
R1:-1.0+i1.4142135623730951 R2:-1.0-i1.4142135623730951

C:\Users\bmsce\Desktop\1BM21CS047\week-1>
```



```
Microsoft Windows [Version 10.0.19044.2251]
(c) Microsoft Corporation. All rights reserved.

C:\Users\student>cd C:\Users\student\Desktop\ooj 1BM21Cs037

C:\Users\student\Desktop\ooj 1BM21Cs037>javac solution.java

C:\Users\student\Desktop\ooj 1BM21Cs037>java solution.java
Enter the values of a,b,c:
0 1 2
Invalid Input

C:\Users\student\Desktop\ooj 1BM21Cs037>java solution.java
Enter the values of a,b,c:
1 2 1
Roots are real and equal
roots are : -1.0

C:\Users\student\Desktop\ooj 1BM21Cs037>java solution.java
Enter the values of a,b,c:
1 2 3
Roots are imaginary

C:\Users\student\Desktop\ooj 1BM21Cs037>java solution.java
Enter the values of a,b,c:
2 5 2
Roots are real and distinct
r1 : -0.5r2 : -2.0

C:\Users\student\Desktop\ooj 1BM21Cs037>
```

Week 2

2. 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.

2/12/22 Week 2

2) Develop a Java Program to create a class Student with members USN, name, an array Credits & marks. Include methods to accept & display details & a method to calculate cgra of student.

```

import java.util.Scanner;

class Student {
    String USN, Name;
    int []marks = new int [5];
    int []Credits = new Credits [5];

    void accept() {
        Scanner SS = new Scanner(System.in);
        System.out.println("Enter USN");
        USN = SS.nextLine();
        System.out.println("Enter Name");
        Name = SS.nextLine();
        for(int i=0; i<5; i++) {
            marks[i] = SS.nextInt();
        }
        for(int i=0; i<5; i++) {
            Credits[i] = SS.nextInt();
        }
    }

    void display() {
        System.out.println("USN : " + USN);
        System.out.println("Name : " + Name);
        System.out.println("Marks in 5 Subjects ");
    }
}

```

```

for (int i=0; i<5; i++) {
    cout << "Marks[" << i << "]";
    cout << endl;
}

cout << endl;
cout << "GPA of 5 subjects";
for (int i=0; i<5; i++) {
    cout << marks[i];
    cout << " ";
}

cout << endl;
cout << "Total GPA";
float gpa() {
    float sum = 0;
    for (int i=0; i<5; i++) {
        sum += marks[i];
    }
    float avg = sum / 5;
    return avg;
}

class Test {
public:
    static void main() {
        Student s1 = new Student();
        s1.accept();
        s1.display();
        float f = s1.gpa();
        cout << f;
    }
}

```

```

int grade (float marks) {
    if (marks > 89)
        return 10;
    else if (marks > 79)
        return 9;
    else if (marks > 69)
        return 8;
    else if (marks > 59)
        return 7;
    else if (marks > 39)
        return 6;
    else
        return 0;
}

```

Output.

Enter your name : 1bm21CS02
 Enter credit & marks of each subject respectively

1	55
3	65
2	75

Total Credit = 9

SGPA = ~~5.77~~ 7.77

The screenshot shows a Windows desktop environment. In the center, there is a Command Prompt window titled "Command Prompt" with the following text:

```
C:\Users\student\Desktop\oop_1BM21Cs037\week_2>javac lab2.java
C:\Users\student\Desktop\oop_1BM21Cs037\week_2>java lab2
Enter your Name
surag
Enter USN
ibm21cs037
Enter credits and marks of each subject respectively
4 55
3 65
for(int i=0;i<3;i++)
    System.out.print("credits Marks
System.out.print("Name:surag USN:ibm21cs037
Total credits=9
SGPA=6.777777777777777
C:\Users\student\Desktop\oop_1BM21Cs037\week_2>
```

To the left of the Command Prompt window, there is a Notepad window titled "lab2.java - Notepad" containing the following Java code:

```
lab2.java - Notepad
File Edit Format View Help
if(marks[i]>=60 && marks[i]<70)
    result+=7*credits[i];
if(marks[i]>=50 && marks[i]<60)
    result+=6*credits[i];
if(marks[i]>=40 && marks[i]<50)
    result+=5*credits[i];
else result+=0*credits[i];
}
for(int i=0;i<3;i++)
    total+=credits[i];
result=result/total;
}
void display()
{
    System.out.print("Enter credits and marks of each subject respectively
    System.out.print("4 55
    System.out.print("3 65
    for(int i=0;i<3;i++)
        System.out.print("credits Marks
        System.out.print("Name:surag USN:ibm21cs037
    Total credits=9
    SGPA=6.777777777777777
}
class lab2
{
    public static void main(String[])
    {
        student s=new student();
        s.accept();
        s.calculate();
        s.display();
    }
}
```

At the bottom of the screen is a dark-colored taskbar with various icons and status information. From left to right, it includes: the Windows logo, a search bar with placeholder text "Type here to search", pinned application icons for File Explorer, Mail, Photos, and Google Chrome, a system clock showing "11:47 AM", a date and time stamp "12/2/2022", and weather information "24°C Partly sunny".

Week 3

3. Create a class Book which contains four members: name, author, price, num_pages. Include a constructor to set the values for the members. Include methods to set and get the details of the objects. Include a `toString()` method that could display the complete details of the book. Develop a Java program to create n book objects.

week-3

3) Create a class Book which contains four members, name, author, price, no pages. Include a Constructor to set the values for members. Include methods to set and get the details of the object. Include `toString()` method that could display the complete details of the book. Develop a Java program to create a book object.

```
import java.util.Scanner;  
  
class Book {  
    String name;  
    String author;  
    int pages;  
    double price;  
  
    Book() {}  
    Scanner SS = new Scanner(System.in);  
  
    void set() {  
        System.out.println("Enter Book name : ");  
        name = SS.nextLine();  
        System.out.println("Enter Author name : ");  
        author = SS.nextLine();  
        System.out.println("Enter number of Pages : ");  
        pages = SS.nextInt();  
        System.out.println("Enter price of the book ");  
        price = SS.nextDouble();  
    }  
  
    public String toString() {  
        return ("Name : " + name + "\nAuthor : " + author +  
               "\nNumber of Pages : " + pages + "\nPrice : " +  
               price);  
    }  
}
```

```

class Lib {
    public static void main (String [] args) {
        Scanner ss = new Scanner (System.in);
        int n;
        System.out.println ("Enter the number of books : ");
        n = ss.nextInt ();
        Book B [] = new Book [n];
        for (int i = 0; i < n; i++) {
            B [i] = new Book ();
            B [i].setC ();
            System.out.println (B [i]);
        }
    }
}

```

Output

Enter the number of books :
 2
 Enter Book name : Doctor
 001_observation
 Enter the price of the book :
 300
 Enter the number of pages :
 100
 Name : 001_observation
 author : B.C. Surya
 price : 300
 Pages : 100

OUTPUTp

The screenshot shows a Windows desktop environment. In the center, there is a window titled "labjava - Notepad" containing Java code. Below the code, a command prompt window is open, showing the execution of the Java program. The taskbar at the bottom has several icons, including the Start button, a search bar, and pinned applications like File Explorer, Edge, and Mail.

```
void set(){
    System.out.println("Enter Book name : ");
    name = SS.nextLine();
    System.out.println("Enter Author name : ");
    author = SS.nextLine();
    System.out.println("Enter the number of pages : ");
    pages = SS.nextInt();
    System.out.println("Enter the price of the Book : ");
    price = SS.nextDouble();
}
class lab3{
    public static void main(String[] args) {
        Scanner SS = new Scanner(System.in);
        int n;
        System.out.print("Enter Number of pages : ");
        n = SS.nextInt();
        book B[] = new book[n];
        for(int i=0; i<n; i++){
            System.out.print("Enter Book name : ");
            name = SS.nextLine();
            System.out.print("Enter Author name : ");
            author = SS.nextLine();
            System.out.print("Enter the number of pages : ");
            pages = SS.nextInt();
            System.out.print("Enter the price of the Book : ");
            price = SS.nextDouble();
            B[i] = new book(name, author, pages, price);
        }
    }
}
```

```
C:\Users\student\Desktop\oop_1BM21Cs037\week_3>java lab3
Enter Book name :
O_O_textBook
Enter Author name :
B C Surag
Enter the number of pages :
1000
Enter the price of the Book :
3000
Author : B C Surag
Book name : O_O_textBook
Author : B C Surag
Enter Number of pages : 1000
The price is : 2000.0
C:\Users\student\Desktop\oop_1BM21Cs037\week_3>
```

Week 4

```
import java.util.Scanner;
abstract class Shape
{
Shape(){}
int height,length;
abstract void printArea();
}
class Rectangle extends Shape
{
Scanner S=new Scanner(System.in);
void printArea()
{
System.out.println("Enter height and width of rectangle");
height=S.nextInt();
length=S.nextInt();
System.out.println("Area of Rectangle is "+length*height);
}
Rectangle(){}
}

class Triangle extends Shape
{
Scanner S=new Scanner(System.in);
void printArea()
{
System.out.println("Enter height and base of triangle");
height=S.nextInt();
length=S.nextInt();
System.out.println("Area of Trianle is "+0.5*length*height);
}
Triangle(){}
}

class Circle extends Shape
{
Scanner S=new Scanner(System.in);
void printArea()
{
System.out.println("Enter radius of Circle");
height=S.nextInt();
System.out.println("Area of Circle is "+3.14*height*height);
}
```

```
Circle(){}  
}  
  
class main  
{  
public static void main(String xx[])  
{  
    Rectangle r=new Rectangle();  
    r.printArea();  
    Triangle t=new Triangle();  
    t.printArea();  
    Circle c=new Circle();  
    c.printArea();  
}  
}
```

Ques-h

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

import java.util.Scanner;
abstract class Shape

```
{  
    Shape() {}  
    int height, length;  
    abstract void printArea();  
}
```

class Rectangle extends Shape

```
{  
    Scanner S = new Scanner(System.in);  
    void printArea()  
    {  
        System.out.println("Enter height & width of rectangle");  
        height = S.nextInt();  
        length = S.nextInt();  
        System.out.println("Area of rectangle is " + length * height);  
    }  
}
```

~~Rectangle()~~

Class Triangle extends Shape

{ Scanner S-newScanner (System.out);

void printArea()

{ System.out.println ("Enter height & base of triangle");

height = S.nextInt();

length = S.nextInt();

System.out.println ("Area of Triangle is " + 0.5 * height * length);

height = S.nextInt();

length = S.nextInt();

System.out.println ("Area of Triangle is " + 0.5 * height * length);

Triangle {}

}

Class Circle extends Shape {

{ Scanner S-newScanner (System.out);

void printArea()

{ System.out.println ("Enter radius of Circle ");

radius = S.nextInt();

System.out.println ("Area of Circle is " + 3.14 * radius * radius);

Circle {}

2-obj

class Main

{

 public static void main (String [] args) {
 System.out.println ("Area of rectangle is " + area);
 System.out.println ("Area of triangle is " + area);
 System.out.println ("Area of circle is " + area);
 }

}

Output

Enter height & width of rectangle

10 20

Area of rectangle is 200

Enter height & base of Triangle

10 20

Area of triangle is 100

~~Enter radius of Circle~~

20

1256

Op: new -> print

area ->

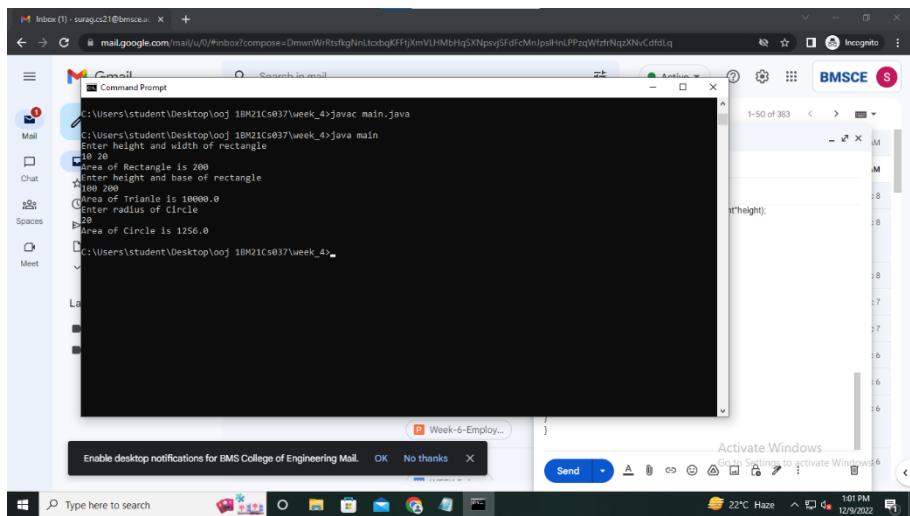
values ->

200.000000

(6.283185307179589)

("rectangle") -> string

new ->



Week 5

```
import java.util.Scanner;
import java.lang.Math;

class Account{

    String name = new String();
    int acc_no;
    double balance;

    void setd(){
        Scanner S = new Scanner(System.in);
        System.out.println("Enter your name : ");
        name = S.nextLine();
        System.out.println("Enter Account number : ");
        acc_no = S.nextInt();
        System.out.println("Enter balance: ");
        balance = S.nextDouble();
    }

    void display(){
        System.out.println("Name : " + name);
        System.out.println("Account number : " + acc_no);
        System.out.println("Balance : " + balance);
    }

    Account(){}
}

class Savings extends Account{
    Scanner S = new Scanner(System.in);
    Savings(){
        System.out.println("Facilities available are : ");
        System.out.println("1.Withdraawal \n 2.Compound Intrest \n 3.No Cheque");
    }

    void deposit(){
        int choice;
        double dep;
        double wd;
        System.out.println("Enter 1 to deposit : ");
        choice = S.nextInt();
        if(choice == 1){
            System.out.println("Enter the amount to deposit : ");
            dep = S.nextDouble();
            balance += dep;
        }
        else
    }
}
```

```

        System.out.println("Invalid Input");
    }
    void intrest()
    {
        System.out.println("Enter rate of interest :");
        double r = S.nextDouble();
        r = r/100;
        System.out.println("Enter frequency of interest applied per time period :");
        int n = S.nextInt();
        System.out.println("Enter time periods :");
        int t = S.nextInt();
        double x = (1+(r/n));
        double compond_intrest = balance*Math.pow(x,n*t);
        System.out.println("Interest amount="+compond_intrest+balance+"\nBalance
amount without interest is"+balance);
        balance = compond_intrest;
        System.out.println("Available balance after updating is : "+balance);
    }
    void withdraw(){
        double wd;
        int choice;
        System.out.println("Enter 1 to withdraw :");
        choice = S.nextInt();
        if(choice == 1){
            System.out.println("Enter the amount you want :");
            wd = S.nextDouble();
            if(wd < balance){
                balance = balance - wd;
                System.out.println("Avilable balance is : " + balance);
            }
            else
                System.out.println("Insufficient balance");
        }
        else
            System.out.println("Invalid Input");
    }
}

class Current extends Account{
    Scanner S = new Scanner(System.in);
    Current()
    {
        System.out.println("Cheque Facility available ");

    }
    void deposit()
    {
        int choice;

```

```

        double amount;
        System.out.println("Press 1 to deposit ");
        choice = S.nextInt();
        if(choice==1)
        {
            System.out.println("Enter amount to be deposited ");
            amount=S.nextDouble();
            balance += amount;
        }
        else
            System.out.println("Invalid Input");
    }

    void withdraw()
    {
        System.out.println("Press 1 to withdraw ammount");
        int choice=S.nextInt();
        if(choice==1)
        {
            System.out.println("Enter the amount to be withdrawn ");
            double wd=S.nextDouble();
            balance = balance - wd;
            System.out.println("Available Balance:"+balance);
        }
        else
            System.out.println("Invalid input");

        if(balance<1000)
        {
            System.out.println("You are running out of minimum balance \nAmount of rs
500 has been debited as service charge for having low balance");
            balance =balance - 500;
            System.out.println("Your Available Balance:"+balance);
        }
    }
}

class Lab5
{
    public static void main(String xx[])
    {
        Scanner S = new Scanner(System.in);
        int choice;
        System.out.println("\nPress\n 1. for Savings account \n2.for Current account");
        choice = S.nextInt();
        switch(choice)
        {
            case 1:

```

```
Savings s1=new Savings();
s1.setd();
s1.display();
s1.deposit();
s1.intrest();
s1.withdraw();
break;
case 2:
    Current c1=new Current();
    c1.setd();
    c1.display();
    c1.deposit();
    c1.withdraw();
    break;
default : System.exit(0);
}
}
}
```

Lab-5

→ Develop a Java program to create a Clear Bank that maintains two types of account for its customer, one Savings Account. Savings provide Compound interest, withdrawal but no Cheque. Current provides Cheque & but no interest. Current account should maintain minimum balance by default, below which a charge is imposed.

Create a class Account with name Accno. Then derive Current & Savings. Proper

- 1) Accept deposit
- 2) Display Balance
- 3) Compute interest
- 4) Permit withdrawal & update Balance
- 5) Check min balance

import java.util.Scanner;
import java.util.Maths;

Clear Account

```
String name = new String();  
int accno;  
double balance;  
void add() {  
    Scanner S = new Scanner(System.in);  
    System.out.println("Enter name :");  
    name = S.nextLine();  
}
```

```
System.out.println("Enter account no:");  
accno = S.nextInt();  
System.out.println("Balance: " + balance);  
balance = S.nextInt();
```

{

```
void display()
```

```
System.out.println("Name: " + name);  
System.out.println("Account No: " + accno);  
System.out.println("Balance: " + balance);
```

}

```
Account()
```

}

```
class Savings extends Account {
```

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

```
    Savings()
```

```
    System.out.println("Facilities available are: ");
```

```
    System.out.println("1. withdraw 2. deposit  
    3. quit")
```

```
    void deposit()
```

```
        int choice;
```

```
        double dep;
```

```
        double bal;
```

```
        System.out.println("Enter 1 to deposit? ");
```

```
        choice = S.nextInt();
```

```
        if (choice == 1) {
```

System.out.println("Enter the amount to deposit: ");

dep = s.nextInt();

balance += dep;

}

else

System.out.println("Invalid input");

}

void insert()

{

System.out.println("Enter interest rate");

double r = s.nextDouble();

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

int t = s.nextInt();

double x = (1 + (r / 100));

double compoundInterest = balance * Math.pow(x, t);

System.out.println("Enter " + compoundInterest + "

Balance " + balance);

balance = compoundInterest;

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

y

void withdraw()

double wd;

System.out.

System.out.println("Enter amount to withdraw: ");

choice = s.nextInt();

void withdraw()

{ System.out.println("Please type withdrawal");
choice = S.nextInt();

if (choice == 1)

{ System.out.println("Please type withdrawal amount");

double wd = S.nextInt();

balance = balance - wd;

System.out.println("Balance: " + Balance);

else

System.out.println("Invalid");

if (balance < 1000)

{ System.out.println("Low Balance Service Fee");

balance = balance - 500;

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

}

}

close

$\{$ (else == 1) {

System.out.println("Data amount");
 val = S.netDouble();

$\{$ (odd balanced)

 balance = balance - val;

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

$\}$

else

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

$\}$

class Correct extends Account {

 Scanner S = new Scanner(System.in);

 double

 { System.out.println("Cheque Facility available");

$\}$

 void deposit() {

 int choice;

 double amount;

 System.out.println("Enter amount to withdraw, to deposit");

 amount = S.netDouble();

 balance += amount;

$\}$

 else

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

$\}$

class dab5{

public static void main(String[] args){

Scanner S = new Scanner(System.in);

put choice

choice = S.nextInt();

switch(choice){

Case 1:

Savings S1 = new Savings();

S1.setd(0);

S1.display();

S1.deposit(5);

S1.withdraw(1);

S1.withdraw(1);

break;

Case 2:

Credit C1 = new Credit();

C1.setdc();

C1.display();

C1.deposit(0);

C1.withdraw(0);

break;

default: System.exit(0);

}

. write

Output

Enter

1. For Savings Account
2. For Current Account.

1
Facilities available are:

1. Withdrawal
2. Compound interest

Enter your name:

Surya

Enter Account no:

123

Enter Balance

1000

Name: Surya

Acno: 123

Balance: 1000

Enter 1 to deposit +

1

Enter Amount

1000

Enter rate of interest

5

Enter frequency of interest

1

Interest amount = 100.0

Balance amount without interest 2000.0

Available balance after interest = 2100.0

part 1 to withdraw:

1

Enter amount

100

Available balance is : 2000

~~09/12/2021~~

```
Command Prompt
C:\Users\student\Desktop\oop\IBN21Cs037\week_5>javac lab5.java
C:\Users\student\Desktop\oop\IBN21Cs037\week_5>java Lab5

Press
1. for Savings account
2. for Current account
3.
Facilities available are :
1.Withdrawal
2.Compound Intrest
3.Balance Cheque
Enter your name :
surag
Enter Account number :
123
Enter balance:
1000
Name : surag
Account number : 123
Balance : 1000.0
Enter 1 to deposit :
1
Enter the amount to deposit :
1000
Enter rate of interest :
5
Enter frequency of interest applied per time period :
1
Enter time periods :
1
Interest amount=100.0
Balance amount without interest is2000.0
Available balance after updating is : 2100.0
Enter 1 to withdraw :
1
Enter the amount you want :
100
Available balance is : 2000.0
C:\Users\student\Desktop\oop\IBN21Cs037\week_5>
```

Activate Windows
Go to Settings to activate Windows.

22°C Haze 1:05 PM 13/9/2022

Week 6

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 cases both father and son's age and throws an exception if son's age is >=father's age.

```
import java.util.Scanner;
class WrongAgeException extends Exception{
    public String toString(){
        return ("Age cannot be negative");
    }
}

class AgeException extends Exception{
    public String toString(){
        return("Age of son cannot be greater than Father's age");
    }
}

class Father{
    int father_age;
    Father(int x) throws WrongAgeException{
        father_age=x;
        if(father_age<0){
            throw new WrongAgeException();
        }
    }
}

class Son extends Father{
    int son_age;
    Son(int x,int y) throws AgeException, WrongAgeException{
        super(x);
        son_age=y;
        if(son_age<0){
            throw new WrongAgeException();
        }
        if(son_age>=father_age){
            throw new AgeException();
        }
    }
}
```

```

        }
    }

class Lab_7{
    public static void main(String[] args) {
        try {
            Scanner s=new Scanner(System.in);
            System.out.println("Enter father's age and Son's age : ");
            int x=s.nextInt();
            int y=s.nextInt();
            Son so=new Son(x,y);
            System.out.println("Father is " + so.father_age + " years old and son is " +
so.son_age + " years old");
        }
        catch (WrongAgeException wa) {
            System.out.println(wa);
        }
        catch (AgeException a){
            System.out.println(a);
        }
        catch (Exception e){
            System.out.println("Enter valid values of age");
        }
    }
}

```

Output

Command Prompt

```

C:\Users\student\Desktop\ooj_1BM21Cs037\week_6>java Lab_7
Enter father's age and Son's age :
10 100
Age of son cannot be greater than Father's age

C:\Users\student\Desktop\ooj_1BM21Cs037\week_6>java Lab_7
Enter father's age and Son's age :
-10 -5
Age cannot be negative

```

```

Enter father's age and Son's age :
100 50
Father is 100 years old and son is 50 years old

```

Observation

lab 6

Write a Java program that demonstrates handling exceptions via inheritance. Create a base class called 'father' and derived class called "Son" which extends the base class. In Father class, implemented a constructor which takes the age & throws the exception WrongAge() when input age < 0. In Son class, implemented a constructor that calls both father & reads age & throws an exception if son's age is \geq father's age.

import java.util.Scanner;

class WrongAgeException extends Exception {
 public WrongAgeException() {
 System.out.println("Age cannot be negative");
 }
}

class AgeException extends Exception {
 public WrongAgeException() {
 System.out.println("Age of Son cannot be greater than Father's age");
 }
}

class Father {
 int fatherAge;
 Father(int x) throws WrongAgeException {
 fatherAge = x;
 if (fatherAge < 0) {
 throw new WrongAgeException();
 }
 }
}

class Son extends Father {
 int sonAge;
 Son(int x) {
 super(x);
 sonAge = x;
 if (sonAge >= fatherAge) {
 throw new WrongAgeException();
 }
 }
}

handle
base class
Son "繼承"
not a
he except
base, implemented
age
= father's age

class Son extends Father {
 int sonAge;
 Son (int x, int y) throws AgeException {
 super (x);
 sonAge = y;
 if (sonAge < 0) {
 throw new AgeException();
 }
 if (sonAge >= fatherAge) {
 throw new AgeException();
 }
 }
}

class Father {
 public static void main (String [] args) {
 try {
 Scanner S = new Scanner (System.in);
 System.out.println ("Enter father's age: ");
 int x = S.nextInt();
 int y = S.nextInt();
 Son so = new Son (x, y);
 System.out.println ("Father's age is " + so.fatherAge +
 " years old and son's age is " + so.sonAge + " years");
 }
 }
}

```
catch (Age age, Exception ex) {
    System.out.println("a");
}
```

```
} catch (AgeException ae) {
    System.out.println("a");
}
```

```
} catch (Exception e) {
    System.out.println("e");
}
```

Output

Enter father's age and son's age:

10 100

Age of son cannot be greater than father's age

Enter father's age and son's age:

-10 -5

Age cannot be negative

Enter father's age and son's age:

100 50

Father is 100 years old and son is 50 year old.

~~Quan
B - 12 - 2 n~~

WEEK 7

Q)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 Call implements Runnable
{
    int x,time;
    String a;
    Thread t;
    Call(String thread_name,int time1,int ex)
    {
        a=thread_name;
        x=ex;
        time=time1;
        t=new Thread(this,a);
        t.start();
    }
    public void run()
    {
        try{
            for(int j=0;j<x   ;j++)
            {
                System.out.println(a);
                Thread.sleep(time);
            }
        }
        catch(InterruptedException ie)
        {
            System.out.println("Interrupted ");
        }
    }
}

class Lab8
{
    public static void main(String xx[])
    {
        new Call("BMS College of Engineering",10000,2);
        new Call("CSE",2000,10);
    }
}
```

OUTPUT

The screenshot shows a Windows desktop environment. In the center, there is a Command Prompt window titled "Command Prompt" with the following text:

```
C:\Users\student\Desktop\oop_1BM21Cs037\week_7>javac Lab8.java
C:\Users\student\Desktop\oop_1BM21Cs037\week_7>java Lab8
BMS College of Engineering
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
System.out.print
Thread.sleep(tim
)
}
catch(InterruptedException ie)
{
    System.out.println("Inte
")
}
}

class Lab8
{
public static void main(String xx[])
{
    new Call("BMS College of Engineering",10000,2);
    new Call("CSE",2000,10);
}
```

To the left of the Command Prompt window is a Notepad window titled "Lab8.java - Notepad" containing the same Java code. The code defines a class "Call" that implements the Runnable interface. It takes a string name and an integer time as parameters. The run() method creates a new Thread and starts it. The thread prints its name and sleeps for the specified time. The main() method creates two instances of Call, one for "BMS College of Engineering" with a time of 10000 and another for "CSE" with a time of 2000.

The taskbar at the bottom of the screen includes icons for File Explorer, Edge browser, Mail, and Task View, along with a search bar and system status indicators for weather (24°C Haze), battery level (4%), and date/time (11:58 AM, 1/6/2023).

Observation

week 7

wrote 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 Call implements Runnable {  
    int x, time;  
    String a;  
    Thread t;  
    call(String thread_name, int time1, int ex) {  
        a = thread_name;  
        x = ex;  
        time = time1;  
        t = new Thread(this);  
        t.start();  
    }  
    public void run() {  
        for(int j=0; j<2; j++) {  
            System.out.println(a);  
            Thread.sleep(time);  
        }  
    }  
    catch(InterruptedException e) {  
        System.out.println("Interrupted");  
    }  
}
```

Output:

BMS College of Engineering
CSE
CSF

```
class Data8 {  
    public static void main (String xx[]) {  
        new Obj ("BMS College of Engineering", 10000,  
            new Obj ("CSE", 8000, 10));  
    }  
}
```

Output

BMS College of Engineering

CSE
CSF

CSE
CSF

CSE
CSF

CSE
BMS College of Engineering

CSE
CSF

CSE
CSE

CSE
CSF

Week 9

Create a package **CIE** which has two classes- **Student** and **Internals- a subclass of Student**. The class **Student** has members like usn, name, sem. The class **internals** has an array that stores the internal marks scored in five courses of the current semester of the student.

Create another package **SEE** which has the class **External** which is a derived class of **Internals**. This class has an array that stores the **SEE** marks scored in five courses of the current semester of the student. Import the two packages in a file that declares the final marks of n students in all five courses.

File 1

```
package CIE;
```

```
public class Student{  
    public int USN;  
    public String Name;  
    public int Sem;
```

```
    public void setdata(int USN1,String Name1, int Sem1){
```

```
        USN = USN1;
```

```
        Name = Name1;
```

```
        Sem = Sem1;
```

```
}
```

```
    public void getd(){
```

```
        System.out.println("USN is : " + USN + "\nName is : " + Name + "\nSEM : " +  
        Sem);
```

```
}
```

```
}
```

File 2

```
package CIE;
```

```
public class internals extends Student{
```

```

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

public void Setd(int [] m){
    for(int i=0; i<5;i++)
        marks[i] = m[i];
    }

public void getda(){
    for(int i=0; i<5;i++)
        System.out.println("Marks is : " + marks[i]);
    }
}

```

File 3

```

package SEE;
import CIE.*;

public class External extends CIE.internals{
    public int [] Emarks = new int [5];
    public void Setda(int [] m){
        for(int i=0; i<5;i++)
            Emarks[i] = m[i];
    }

    public void getdata(){
        for(int i=0; i<5;i++)
            System.out.println("External Marks is : " + Emarks[i]);
        System.out.println("Final marks is : "+ (Emarks[i]/2 + super.marks[i]))
    }
    }
}

```

File 4

```
import java.util.Scanner;  
import CIE.*;  
import SEE.*;  
  
class Lab_9{  
    public static void main(String args[]){  
        Scanner S = new Scanner(System.in);  
        int marks[] = new int[5];  
        System.out.print("\nEnter the number of students : ");  
        int n = S.nextInt();  
        External [] E = new External[n];  
        System.out.println("Enter Student details ");  
        for(int i=0; i<n; i++){  
            int usn = S.nextInt();  
            String name = S.nextLine();  
            int sem = S.nextInt();  
  
            System.out.println("Enter the internal marks of the student ");  
            for(int j=0;j<5;j++){  
                marks[j] = S.nextInt();  
            }  
            E[i] = new External();  
            E[i].setdata(usn, name, sem);  
            E[i].setd(marks);  
            System.out.println("Enter the External marks of the student ");  
            for(int k=0;k<5;k++){  
                marks[k] = S.nextInt();  
            }  
        }  
    }  
}
```

```
        }  
        E[i].setda(marks);  
        E[i].getd();  
        E[i].getda();  
        E[i].getdata();  
    }  
}
```

OUTPUT

```
Enter number of students
1
Enter student details
1 aqw 2
Enter internal marks
12 12 12 12 12
Enter external marks
12 12 12 12 12
Student details
USN:1
NAME:aqw
SEMESTER:2
Internal marks
Subject 0: 12
Subject 1: 12
Subject 2: 12
Subject 3: 12
Subject 4: 12
External marks
Subject 0: 12
External marks
Subject 1: 12
External marks
Subject 2: 12
External marks
Subject 3: 12
External marks
Subject 4: 12
Final marks
Subject 0: 18
Subject 1: 18
Subject 2: 18
Subject 3: 18
Subject 4: 18
```

OBSERVATION

Week 8

Create 2 packages CSE & SER. CSE with 2 Classes
and Internals SER with interface. Create a list &
queue

File 1

Package CSE;

public class Student {

int USN;

String Name;

int Sem;

public void setdata(int USN, String Name, int Sem) {

USN = USN;

Name = Name;

Sem = Sem;

}

public void getdata() {

System.out.println("USN: " + USN + " Name: " + Name + "\n" + "Sem: " + Sem)

}

8/28
Java
Student
Skills

File 2
Package CJK;

public class Internals extends Student {

 int[] marks = new int [5];

 public void setda (int [] a) {

 for (int i=0; i<5; i++) {

 marks [i] = a[i];

}

 public void getda () {

 for (int i=0; i<5; i++) {

 System.out.println ("Marks of internals " + marks [i]);

 }

File 3 :

```
Package SKT5;
import CIE.*;

public class External extends Intern {
    int Emarks = new int[5];
    public void setdata(int a[]) {
        for (int i=0; i<5; i++) {
            Emarks[i] = a[i];
        }
    }
    public void getdata() {
        for (int i=0; i<5; i++) {
            System.out.println("External marks : " + Emarks[i]);
        }
    }
}
```

File 4

```
import java.util.Scanner;
import CIE.*;
import SKT.*;

class Data {
    public static void main(String args[]) {
        Scanner S = new Scanner(System.in);
        System.out.println("Enter number of Students : ");
        int n = S.nextInt();
        External[] E = new External[n];
    }
}
```

```
for (int i=0; i<n; i++) {
    System.out.println ("Data Student details : ");
    int USN = S.nextInt();
    String name = S.nextLine();
    int Sem = S.nextInt();
```

```
L[9] = new External();
L[9].selld (USN, name, Sem);
```

```
System.out.println ("Enter external marks : ");
int m[] = new int [5];
for (int j=0; j<5; j++)
    m[j] = S.nextInt();
L[9].Setdata (m);
```

```
System.out.println ("Enter External marks : ");
for (int k=0; k<5; k++)
    m[k] = S.nextInt();
```

```
L[9].Setdata ();
```

```
for (int p=0; p<n; p++) {
```

```
L[i].getdc();
L[i].getdc();
L[i].getdc();
```

```
}
```

System.out.println ("Final Marks");
for (int i=0; i<5; i++) {

System.out.println (" " + E[i].marks[0] + E[i].marks[1]);

}

}

Output

Enter number of Students

1

Enter details of Student

1 2

Enter External marks

15 15 15 15 15

Enter Internal marks

30 30 30 30 30

USN: 1

Name b

Sem 3

Internal marks

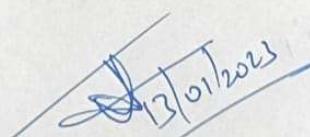
15 15 15 15 15

External marks

30 30 30 30 30

Total marks

30 30 30 30 30


30/01/2023