VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

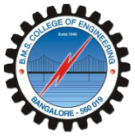
LAB REPORT
on

OBJECT ORIENTED JAVA PROGRAMMING (21CS3PCOOJ)

Submitted by

SPOORTHI J
(1BM21CS218)

in partial fulfillment for the award of the degree of
BACHELOR OF ENGINEERING
in
COMPUTER SCIENCE AND ENGINEERING



B.M.S. COLLEGE OF ENGINEERING
(Autonomous Institution under VTU)
BENGALURU-560019

October-2022 to Feb-2023

B. M. S. College of Engineering,
Bull Temple Road, Bangalore 560019
(Affiliated To Visvesvaraya Technological University, Belgaum)
Department of Computer Science and Engineering

CERTIFICATE

This is to certify that the Lab work entitled "Object Oriented Java Programming (21CS3PCOOJ)" carried out by **Spoorthi J (1BM21CS218),** who is bonafide student of **B. M. S. College of Engineering.** It is in partial fulfillment for the award of **Bachelor of Engineering in Computer Science and Engineering** of the Visvesvaraya Technological University, Belgaum during the year 2022. The Lab report has been approved as it satisfies the academic requirements in respect of a Object Oriented Java Programming (21CS3POOJ) work prescribed for the said degree.

Name of the Lab-Incharge
Designation
Department of CSE
BMSCE, Bengaluru

Dr. Jyothi S Nayak
Professor and Head
Department of CSE
BMSCE, Bengaluru

`

Index

6	05/01/23	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.	**23**
7	12/01/23	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	**26**

Experiment Title

1.Develop a Java program that prints all real solutions to the quadratic equation ax2+bx+c = 0. Read in a, b, c and use the quadratic formula. If the discriminate b2-4ac is negative, display a message stating that there are no real solutions.

```
import java.util.*;
import java.lang.Math.*;
public class Quadratic
{
public static void main(String args[])
{
Scanner in=new Scanner(System.in);
System.out.println("Enter a");
double a=in.nextDouble();
System.out.println("Enter b");
double b=in.nextDouble();
System.out.println("Enter c");
double c=in.nextDouble();
if(a==0)
{
System.out.println("Invalid Inputs \n ");
}
else
{
double d=b*b-4*a*c;
if(d>0.0)
{
double r1=(-b+(Math.sqrt(d)/(2.0*a)));
double r2=(-b-(Math.sqrt(d)/(2.0*a)));
```

```
System.out.println("Roots are real and distinct \n Roots are \n r1="+r1+"\n r2="+r2)
}
else if(d==0.0)
{
double r1=-b/(2*a);
System.out.println("Roots are real and equal and each root is equal to"+r1);
}
else
{
System.out.println("Roots are imaginary and distinct. \n Roots are\n");
double r1=-b/(2.0*a);
double r2=(Math.sqrt(Math.abs(d)))/(2.0*a);
System.out.println("r1= "+r1+"+i"+r2+"\n"+"r2= "+r1+"-i"+r2);
}
}
}
}
```

Output:

```
C:\Users\admin\Desktop\1BM21CS218 JAVA>javac Quadratic.java

C:\Users\admin\Desktop\1BM21CS218 JAVA>java Quadratic
Enter a
0
Enter b
4
Enter c
5
Invalid Inputs


C:\Users\admin\Desktop\1BM21CS218 JAVA>java Quadratic
Enter a
1
Enter b
2
Enter c
3
Roots are imaginary and distinct.
 Roots are

r1= -1.0+i1.4142135623730951
r2= -1.0-i1.4142135623730951

C:\Users\admin\Desktop\1BM21CS218 JAVA>java Quadratic
Enter a
2
Enter b
4
Enter c
2
Roots are real and equal and each root is equal to-1.0
```

Command Prompt

```
C:\Users\admin\Desktop\1BM21CS218 JAVA>java Quadratic
Enter a
1
Enter b
4
Enter c
2
Roots are real and distinct
 Roots are
 r1=-2.585786437626905
 r2=-5.414213562373095
```

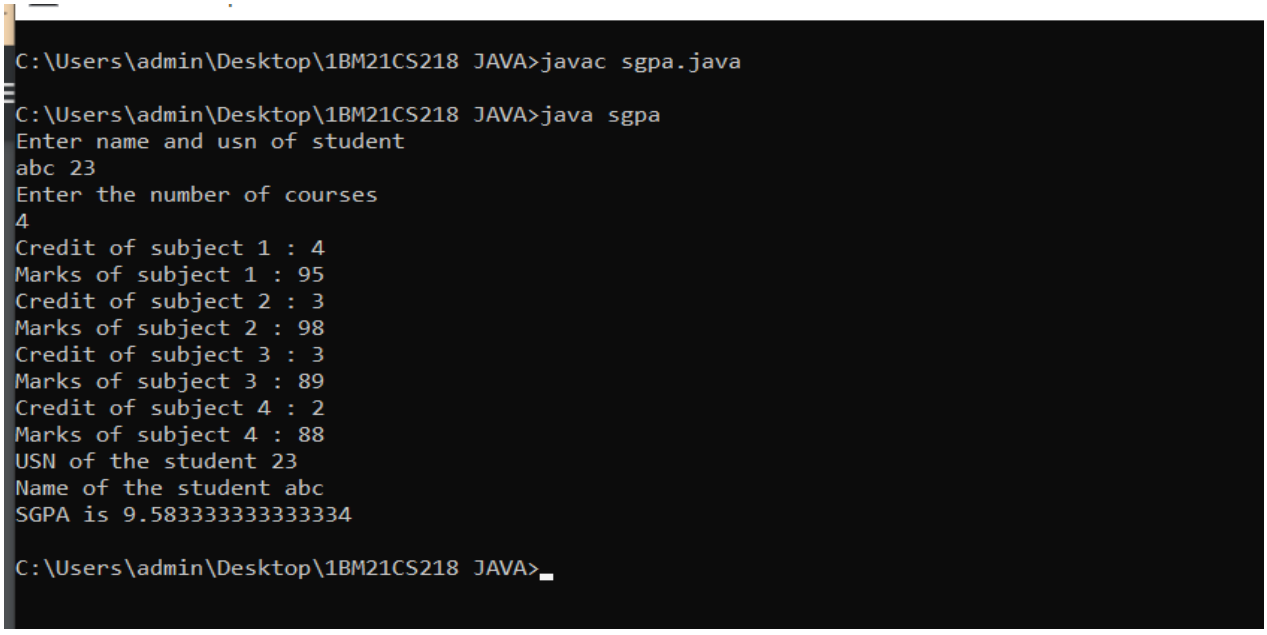
EXPERIMENT-2

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.

```
import java.util.Scanner;
class student{
void display(String name, String usn)
{
System.out.println("USN of the student "+usn);
 System.out.println("Name of the student "+ name);
}
void calculatesgpa(double[] marks, double[] credits, int number)
{
double gradepoints[]=new double[number];
double sgpa,sum=0,tnum=0;
for (int 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++)
{
sum+=credits[i]*gradepoints[i];
}
for(int i=0;i<number;i++)
{
tnum+=credits[i];
}
sgpa=sum/tnum;
System.out.println("SGPA is "+sgpa);
}
```

Output:



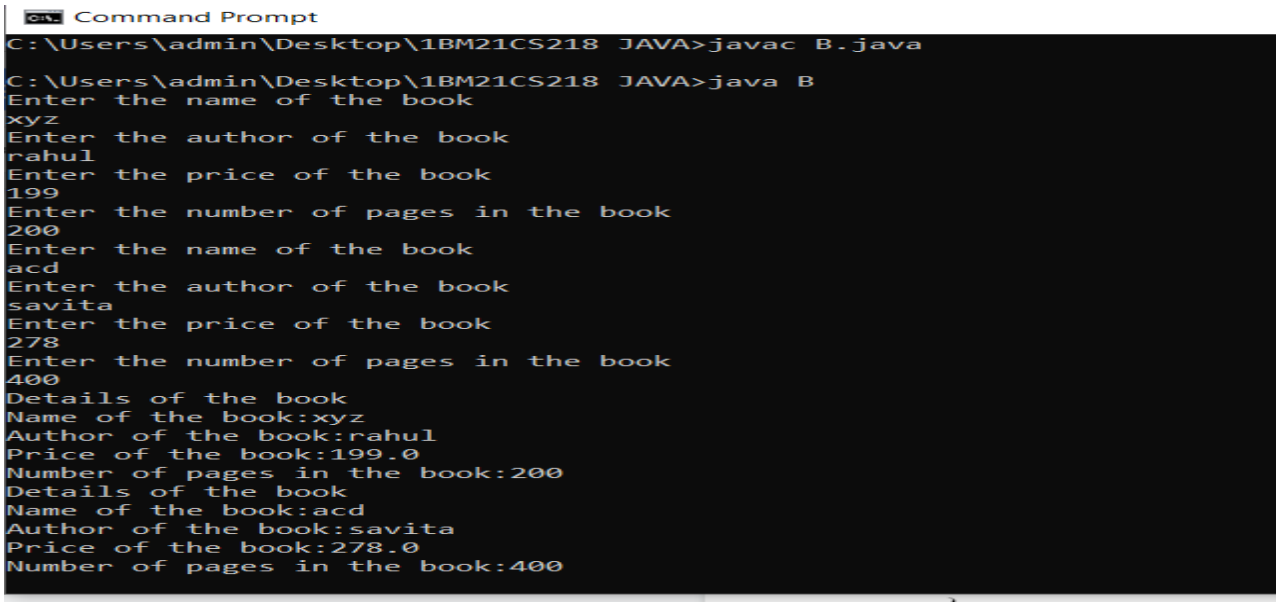
EXPERIMENT-3

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.

```
import java.util.*;
import java.lang.*;
class Book{
String name,author;
double price;
int num_pages;
void getdetails()
{
Scanner b=new Scanner(System.in);
System.out.println("Enter the name of the book");
name=b.next();
System.out.println("Enter the author of the book");
author=b.next();
System.out.println("Enter the price of the book");
price=b.nextDouble();
System.out.println("Enter the number of pages in the book");
num_pages=b.nextInt();
}
void display()
{
System.out.println("Details of the book");
System.out.println("Name of the book:"+name);
System.out.println("Author of the book:"+author);
```

```
System.out.println("Price of the book:"+price);
System.out.println("Number of pages in the book:"+num_pages);
}
}
class B{
public static void main(String args[])
{
Book b1[]=new Book[2];
for(int i=0;i<2;i++)
b1[i]=new Book();
for(int j=0;j<2;j++)
b1[j].getdetails();
for(int k=0;k<2;k++)
b1[k].display();
}
}
```

Output:



EXPERIMENT 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.

```
abstract class Shape{
int a,b;
Shape(int x,int y)
{
 a=x;
 b=y;
}
abstract double printarea();
}
class Rectangle extends Shape{
Rectangle(int x,int y)
{
super(x,y);
}
double printarea()
{
return a*b;
}
}
class Triangle extends Shape{
Triangle(int x,int y)
{
super(x,y);
```

```
}
double printarea()
{
return 0.5*a*b;
}
}
class Circle extends Shape{
Circle(int x)
{
super(x,1);
}
double printarea()
{
return 3.142*a*a;
}
}
class Area{
public static void main(String args[]){
Rectangle r1=new Rectangle(5,10);
Triangle t1=new Triangle(8,6);
Circle c1=new Circle(4);
Shape s;
s=r1;
double a1=s.printarea();
System.out.println("Area of rectangle is:"+a1);
s=t1;
double b1=s.printarea();
System.out.println("Area of triangle is:"+b1);
```

```
s=c1;
double d1=s.printarea();
System.out.println("Area of circle is:"+d1);
}
}
```

Output:

```
C:\Users\admin\Desktop\1BM21CS218 JAVA>javac Area.java

C:\Users\admin\Desktop\1BM21CS218 JAVA>java Area
Area of rectangle is:50.0
Area of triangle is:24.0
Area of circle is:50.272

C:\Users\admin\Desktop\1BM21CS218 JAVA>
```

EXPERIMENT 5

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

a) Accept deposit from customer and update the balance.

b) Display the balance.

c) Compute and deposit interest

d) Permit withdrawal and update the balance Check for the minimum balance, impose penalty if necessary and update the balance

```
import java.util.Scanner;
import java.lang.Math;
class Account
{
String name, acc_type;
int acc_no;
double bal,dep;
Scanner scan= new Scanner(System.in);
void setd()
{
System.out.println("Enter your Name:");
name=scan.next();
System.out.println("Enter your Account Number:");
```

```
acc_no=scan.nextInt();
System.out.println("Enter your Account type: (Savings/Current)");
acc_type=scan.next();
System.out.println("Enter the Bank Balance:");
bal=scan.nextInt();
}
void disp()
{
System.out.println("Name: "+name);
System.out.println("Account Number: "+acc_no);
System.out.println("Account Type: "+acc_type);
System.out.println("Current balance is: "+bal);
}
void deposit()
{
System.out.println("Enter the amount to be deposited:");
dep=scan.nextInt();
bal+=dep;
System.out.println("BALANCE AMOUNT: "+bal);
}
}
class Cur_acct extends Account
{
int penal()
{
double min, pen;
System.out.println("Enter Minimum balance & penalty amount if not followed:");
min=5000; pen=min*0.05;
```

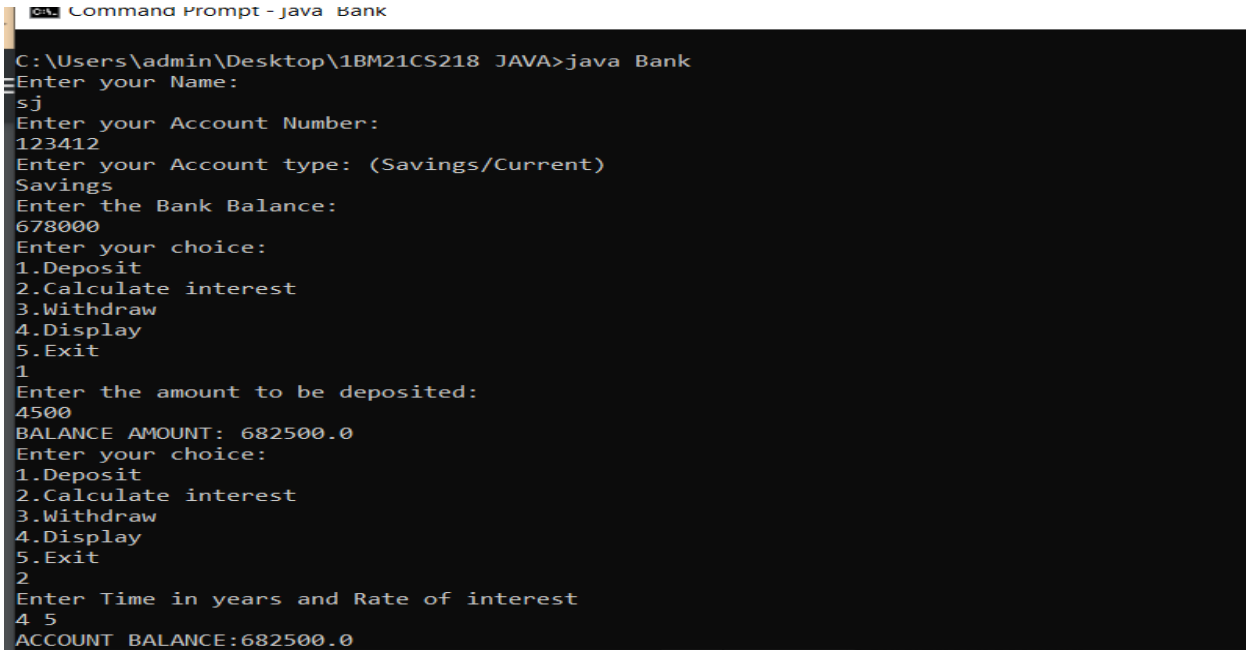
```
if(bal<min)
{
bal-=pen;
System.out.println("Penalty imposed for having insufficient balance");
return 0;
}
else
{
System.out.println("No penalty");
   return 1;
}
}
void withdrawal()
{
double amt;
System.out.println("Enter amount to be withdrawn:");
amt=scan.nextInt();
          int a= penal();
if(a==1)
{
if(bal>=amt)
              { bal=bal-amt;
System.out.println("Account Balance after withdrawal is:" +bal);}
}
else
            System.out.println("The amount can't be withdrawn");
     }
}
```

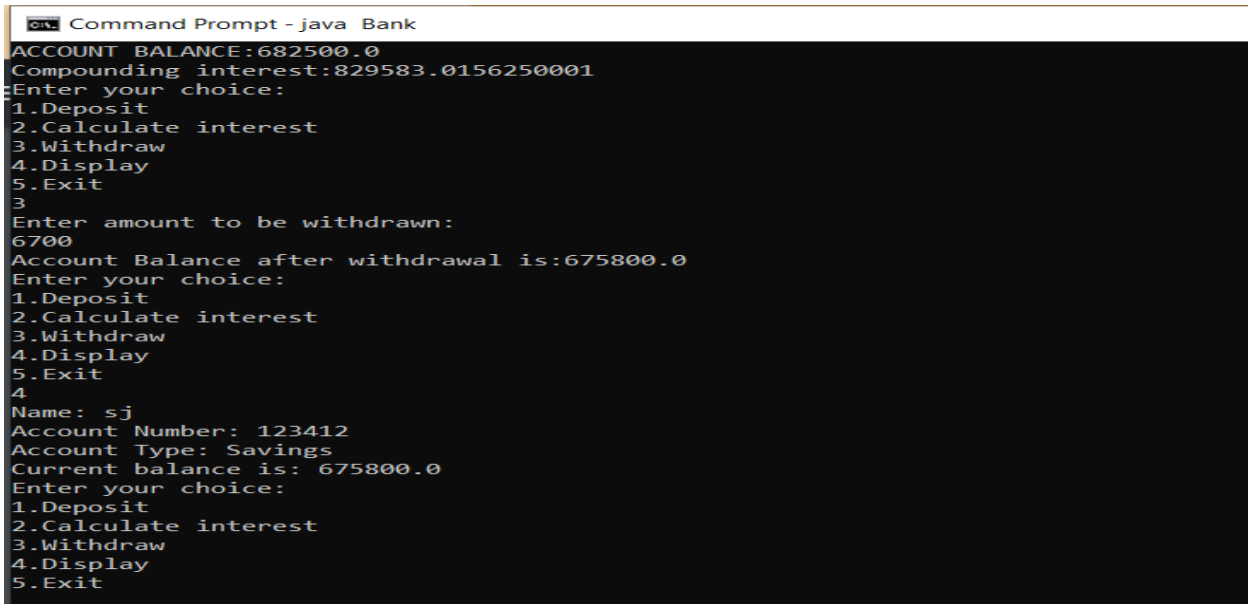
```
class Sav_acct extends Account
{
void calc_interest()
{
System.out.println("Enter Time in years and Rate of interest");
double t=scan.nextDouble(); double r=scan.nextDouble();
double CI = bal*Math.pow((1 + r/ 100), t);
System.out.println("ACCOUNT BALANCE:" + bal);
System.out.println("Compounding interest:"+ CI);
}
void withdrawal()
{
double amt;
System.out.println("Enter amount to be withdrawn:");
amt=scan.nextInt();
if(bal>=amt)
            { bal=bal-amt;
System.out.println("Account Balance after withdrawal is:" +bal);}
else
            System.out.println("The amount can't be withdrawn");
    }
}
class Bank
{
  public static void main(String arg[])
  {
Scanner ss=new Scanner(System.in);
Account b1=new Account();
```

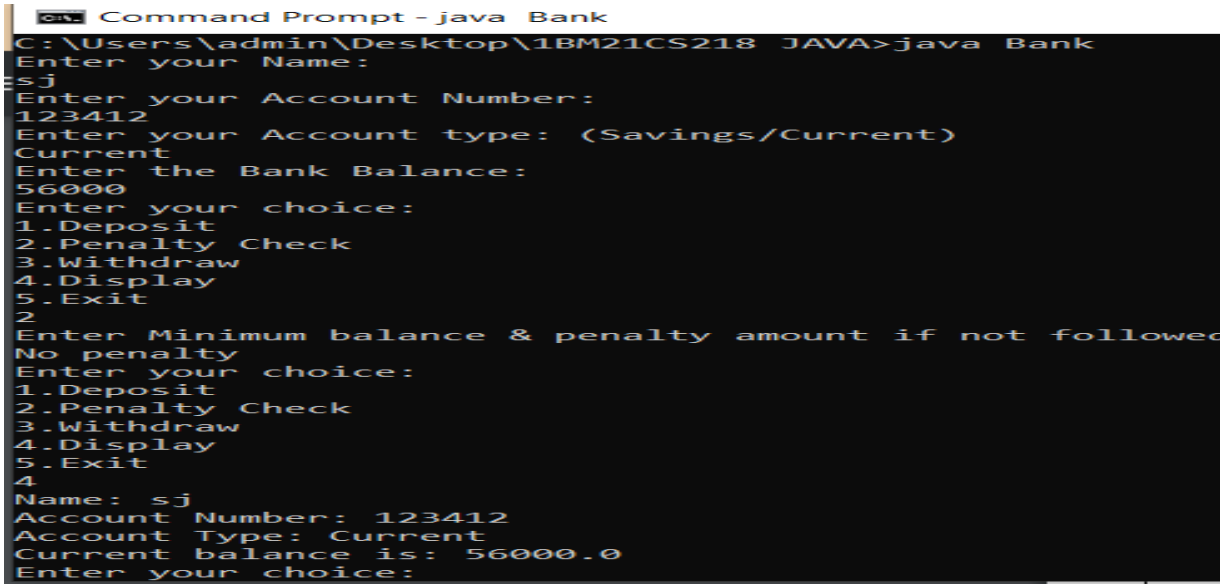
```
b1.setd();
if(b1.acc_type.equals("Savings"))
{
Sav_acct s1=new Sav_acct();
//s1=b1;
s1.name=b1.name; s1.acc_no=b1.acc_no; s1.acc_type=b1.acc_type; s1.bal=b1.bal;
while(true)
{
System.out.println("Enter your choice:\n1.Deposit\n2.Calculate
interest\n3.Withdraw\n4.Display\n5.Exit");
int choice=ss.nextInt();
switch(choice)
{
case 1: s1.deposit(); break;
case 2: s1.calc_interest(); break;
case 3: s1.withdrawal(); break;
case 4: s1.disp(); break;
case 5: System.exit(0);
default: System.out.println("Invalid input");
}
}
}
else if(b1.acc_type.equals("Current"))
{
Cur_acct c1=new Cur_acct();
c1.name=b1.name; c1.acc_no=b1.acc_no; c1.acc_type=b1.acc_type; c1.bal=b1.bal;
while(true)
{
```

```
System.out.println("Enter your choice:\n1.Deposit\n2.Penalty
Check\n3.Withdraw\n4.Display\n5.Exit");
int choice=ss.nextInt();
switch(choice)
{
case 1: c1.deposit(); break;
case 2: c1.penal(); break;
case 3: c1.withdrawal(); break;
case 4: c1.disp(); break;
case 5: System.exit(0);
default: System.out.println("Invalid input");
}
}
}
else
  System.out.println("Invalid Account type");
   }
}
```

Output:


Command Prompt - java Bank
C:\Users\admin\Desktop\1BM21CS218 JAVA>java Bank
Enter your Name:
sj
Enter your Account Number:
123412
Enter your Account type: (Savings/Current)
Savings
Enter the Bank Balance:
678000
Enter your choice:
1.Deposit
2.Calculate interest
3.Withdraw
4.Display
5.Exit
1
Enter the amount to be deposited:
4500
BALANCE AMOUNT: 682500.0
Enter your choice:
1.Deposit
2.Calculate interest
3.Withdraw
4.Display
5.Exit
2
Enter Time in years and Rate of interest
4 5
ACCOUNT BALANCE:682500.0


Command Prompt - java Bank
ACCOUNT BALANCE:682500.0
Compounding interest:829583.0156250001
Enter your choice:
1.Deposit
2.Calculate interest
3.Withdraw
4.Display
5.Exit
3
Enter amount to be withdrawn:
6700
Account Balance after withdrawal is:675800.0
Enter your choice:
1.Deposit
2.Calculate interest
3.Withdraw
4.Display
5.Exit
4
Name: sj
Account Number: 123412
Account Type: Savings
Current balance is: 675800.0
Enter your choice:
1.Deposit
2.Calculate interest
3.Withdraw
4.Display
5.Exit


Command Prompt - java Bank
C:\Users\admin\Desktop\1BM21CS218 JAVA>java Bank
Enter your Name:
sj
Enter your Account Number:
123412
Enter your Account type: (Savings/Current)
Current
Enter the Bank Balance:
56000
Enter your choice:
1.Deposit
2.Penalty Check
3.Withdraw
4.Display
5.Exit
2
Enter Minimum balance & penalty amount if not followed
No penalty
Enter your choice:
1.Deposit
2.Penalty Check
3.Withdraw
4.Display
5.Exit
4
Name: sj
Account Number: 123412
Account Type: Current
Current balance is: 56000.0
Enter your choice:

EXPERIMENT 6

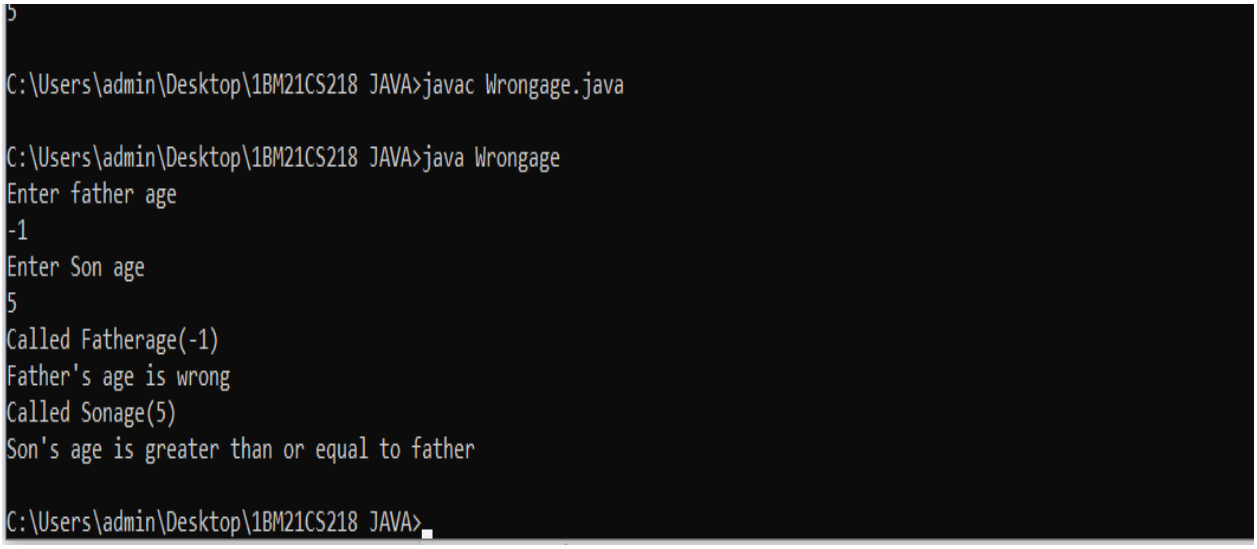
Write a program that demonstrates handling of exceptions in inheritance tree. Create a base class called "Father" and derived class called "Son" which extends the base class. In Father class, implement a constructor which takes the age and throws the exception WrongAge( ) when the input age<0. In Son 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 Father extends Exception{
int fage;
Father(int x)
{
fage=x;
}
public String toString(){
return "Father's age is wrong";
}
}
class Son extends Father{
int sage;
Son(int x,int y)
{
super(x);
sage=y;
}
public String toString(){
return "Son's age is greater than or equal to father";
}
}
```

```
class Wrongage{
static int x,y;
static void Fatherage(int x) throws Father
{
System.out.println("Called Fatherage("+x+")");
if(x<0)
throw new Father(x);
System.out.println("Normal exit father's age is"+x);
}
static void Sonage(int x,int y) throws Son
{
System.out.println("Called Sonage("+y+")");
if(y>=x)
throw new Son(x,y);
System.out.println("Normal exit son's age is"+y);
}
public static void main(String args[])
{
Scanner input=new Scanner(System.in);
System.out.println("Enter father age");
x=input.nextInt();
System.out.println("Enter Son age");
y=input.nextInt();
try{
Fatherage(x);
}
catch(Father e)
{
```

```
System.out.println(e);
}
try{
Sonage(x,y);
}
catch(Son e)
{
System.out.println(e);
}
}
}
```

Output:



EXPERIMENT 7

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
{
String a;
int x,time;
Thread t;
Call(String tn,int ti,int ex)
{
a=tn;
x=ex;
time=ti;
t=new Thread(this,a);
t.start();
}
public void run()
{
try{
for(int i=0;i<x ;i++)
{
System.out.println(a);
Thread.sleep(time);
}
 }
catch(InterruptedException ie)
{
```

```
System.out.println("Inturrupted ");
}
}
}
class sync
{
public static void main(String args[])
{
new Call("BMS College of Enginnering",10000,2);
new Call("CSE",2000,10);
}
}
```

Output: