**VISVESVARAYA TECHNOLOGICAL UNIVERSITY**

**“JnanaSangama”, Belgaum -590014, Karnataka.**

****

**LAB REPORT**

**on**

**OBJECT ORIENTED JAVA**

***Submitted by***

**HARIKA N (1BM21CS071)**

***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**

**Oct 2022-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**” carried out by **HARIKA N(1BM21CS071),** 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 2022-23. The Lab report has been approved as it satisfies the academic requirements in respect of Object oriented java Lab **- (21CS3PCOOJ )**work prescribed for the said degree.

Name of the Lab-Incharge               **Dr. Jyothi S Nayak**

**Vikranth B M**

Assistant Professor Professor and Head

Department of CSE Department of CSE

BMSCE, Bengaluru BMSCE, Bengaluru

`

**Index Sheet**

|  |  |  |
| --- | --- | --- |
| **Sl. No.** | **Experiment Title** | **Page No.** |
| 1 | Quadratic Equation | 4-7 |
| 2 | Calculation of SGPA | 8-11 |
| 3 | Creating n Book Objects | 12-15 |
| 4 | Abstract class named shape | 16-20 |
| 5 | Bank Program | 21-30 |
| 6 | WrongAge exception | 31-34 |
| 7 | Threads program | 35-36 |
|  |  |  |

**Course Outcome**

|  |  |
| --- | --- |
| CO1 | Apply the knowledge of Java concepts to find the solution for a given problem. |
| CO2 | Analyse the given Java application for correctness/functionalities. |
| CO3 | Develop Java programs / applications for a given requirement. |
| CO4 | Conduct practical experiments for demonstrating features of Java. |

**LAB PROGRAM 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.math.\*;

class Dicriminant{

double a,b,c;

Dicriminant(double i,double j,double k)

{

a=i;

b=j;

c=k;

}

double Discri()

{

return ((b\*b)-(4\*a\*c));

}

}

class Quadratic{

public static void main(String args[])

{

Scanner sc=new Scanner(System.in);

double r1,r2,a,b,c;

System.out.println("enter the value of a");

a=sc.nextInt();

System.out.println("enter the value of b");

b=sc.nextInt();

System.out.println("enter the value of c");

c=sc.nextInt();

if (a==0)

{

System.out.println("entered equation is not quadratic");

}

else

{

Dicriminant d=new Dicriminant(a,b,c);

if(d.Discri()>0)

{

r1=(-b+Math.pow(d.Discri(),0.5))/(2\*a);

r2=(-b-Math.pow(d.Discri(),0.5))/(2\*a);

System.out.println("Roots are real and distinct:r1= "+r1 + "r2=" +r2);

}

else if(d.Discri()<0)

{

r1=-b/(2\*a);

r2=(Math.pow(Math.abs(d.Discri()),0.5))/(2\*a);

System.out.println("Roots are complex: r1= "+r1+"+i"+r2+ " r2="+r1+"-i"+r2);

}

else

{

r1=-b/(2\*a);

System.out.println("Roots are real and equal : r1=r2= "+r1);

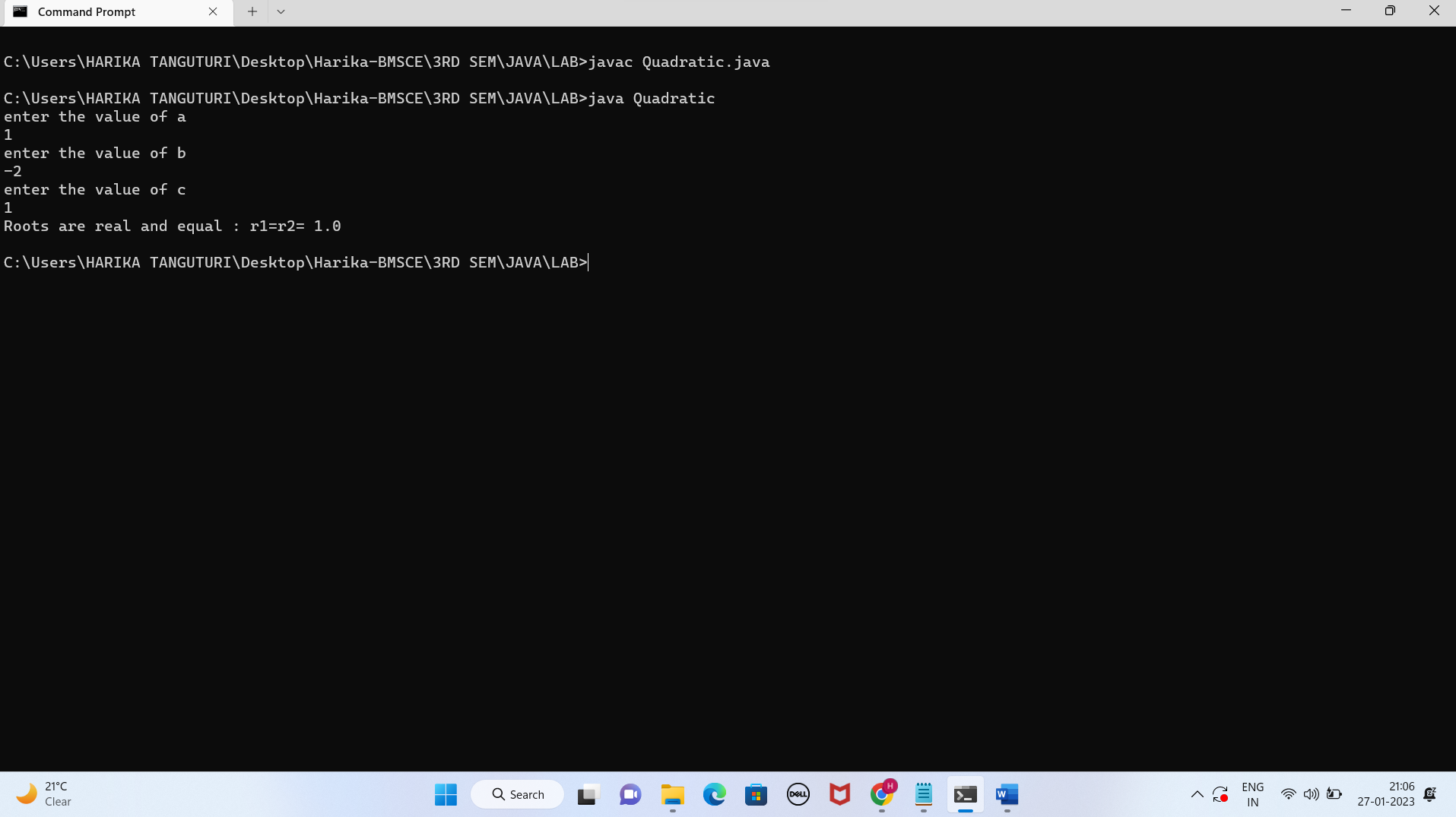
}

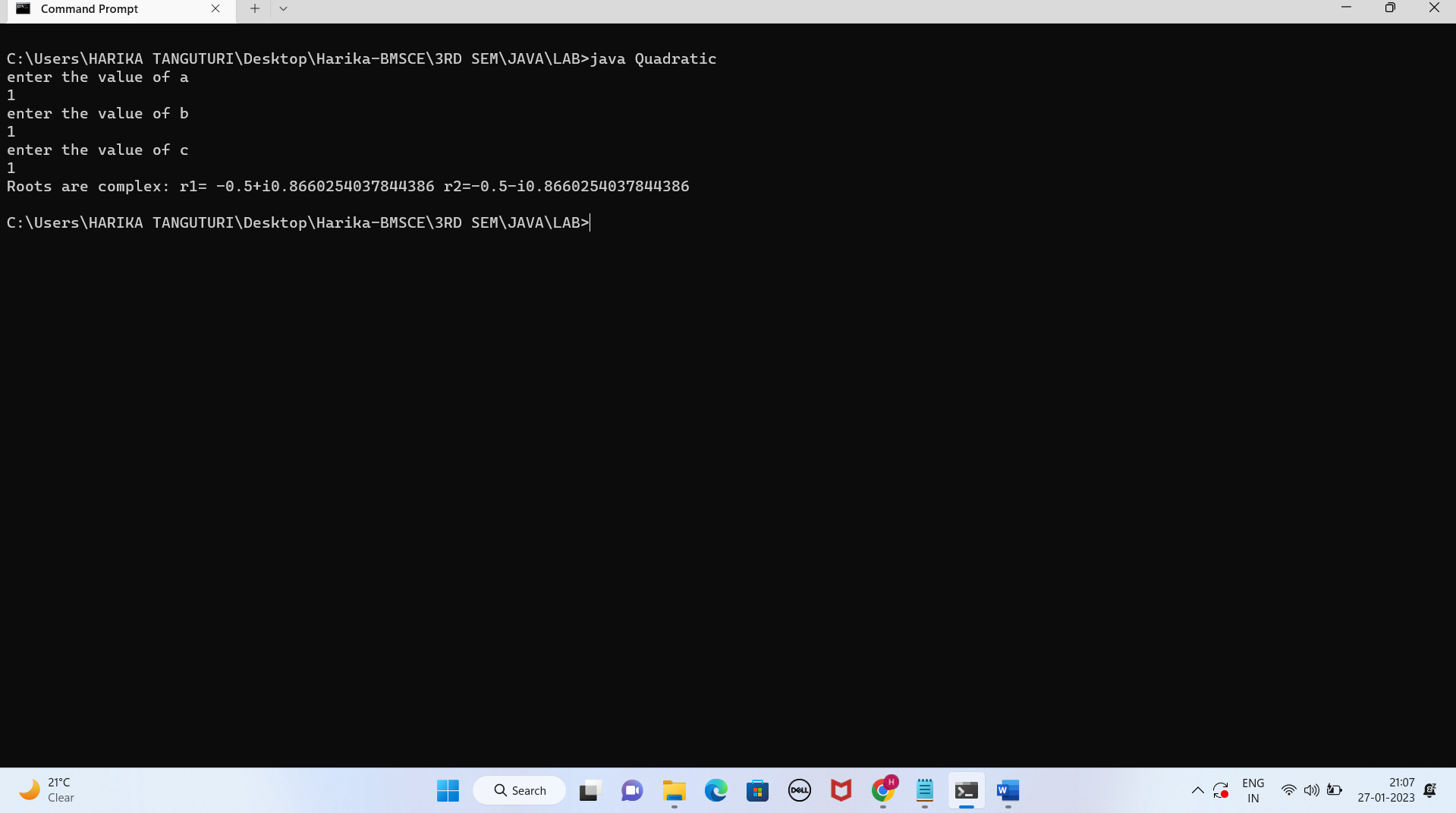
}

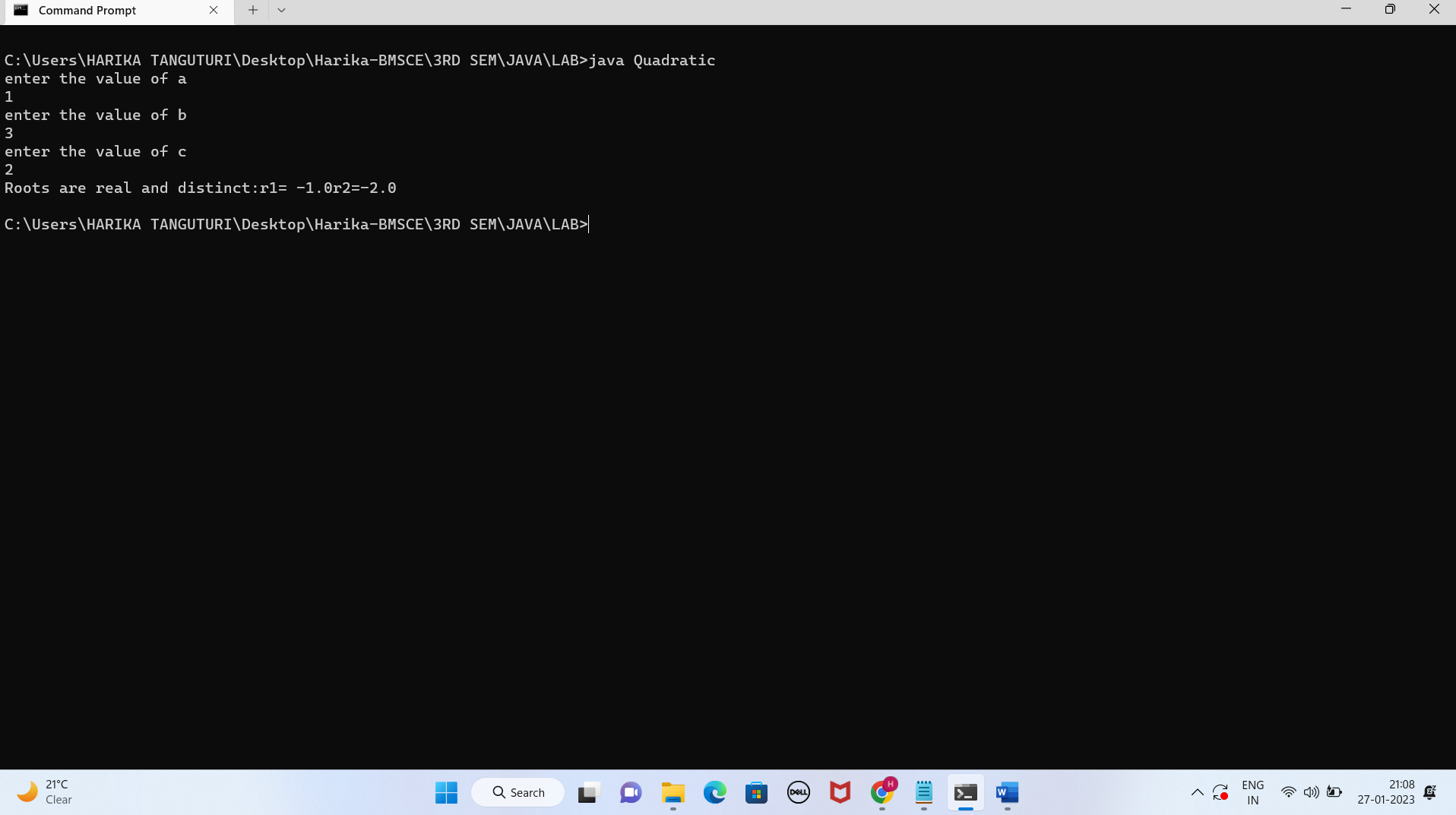
}

}

**OUTPUT:**

****

****

****

**LAB PROGRAM 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.\*;

class Student{

String usn;

String name;

int credits[]=new int[20];

int marks[]=new int[20];

int gradepoints[]=new int[20];

double nume=0;

double denom=0;

double SGPA;

int i,n;

void accept()

{

Scanner sc=new Scanner(System.in);

System.out.println("Enter Student Details");

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

usn=sc.next();

System.out.println("Enter Student Name");

name=sc.next();

System.out.println("Enter number of Subjects");

n=sc.nextInt();

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

{

System.out.println("Enter Subject"+(i+1)+"\t"+"marks");

marks[i]=sc.nextInt();

System.out.println("Enter Subject"+(i+1)+"\t"+"credits");

credits[i]=sc.nextInt();

denom+=credits[i];

}

}

void calculate()

{

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

{

if(marks[i]>=90)

gradepoints[i]=10;

else if(marks[i]>=80 && marks[i]<90)

gradepoints[i]=9;

else if(marks[i]>=70 && marks[i]<80)

gradepoints[i]=8;

else if(marks[i]>=60 && marks[i]<70)

gradepoints[i]=7;

else if(marks[i]>=55 && marks[i]<60)

gradepoints[i]=6;

else if(marks[i]>=50 && marks[i]<55)

gradepoints[i]=5;

else if(marks[i]>=40 && marks[i]<50)

gradepoints[i]=4;

else

gradepoints[i]=4;

nume+=(credits[i]\*gradepoints[i]);

}

SGPA=(nume/denom);

}

void display()

{

System.out.println("The Student Details");

System.out.println("Name: "+name+"\n"+"USN: "+usn);

System.out.println("marks"+"\t"+"credits");

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

{

System.out.println(marks[i]+"\t"+credits[i]);

}

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

}

}

class StudentDemo{

public static void main(String args[])

{

Student s1 = new Student();

Student s2 = new Student();

s1.accept();

s1.calculate();

s1.display();

s2.accept();

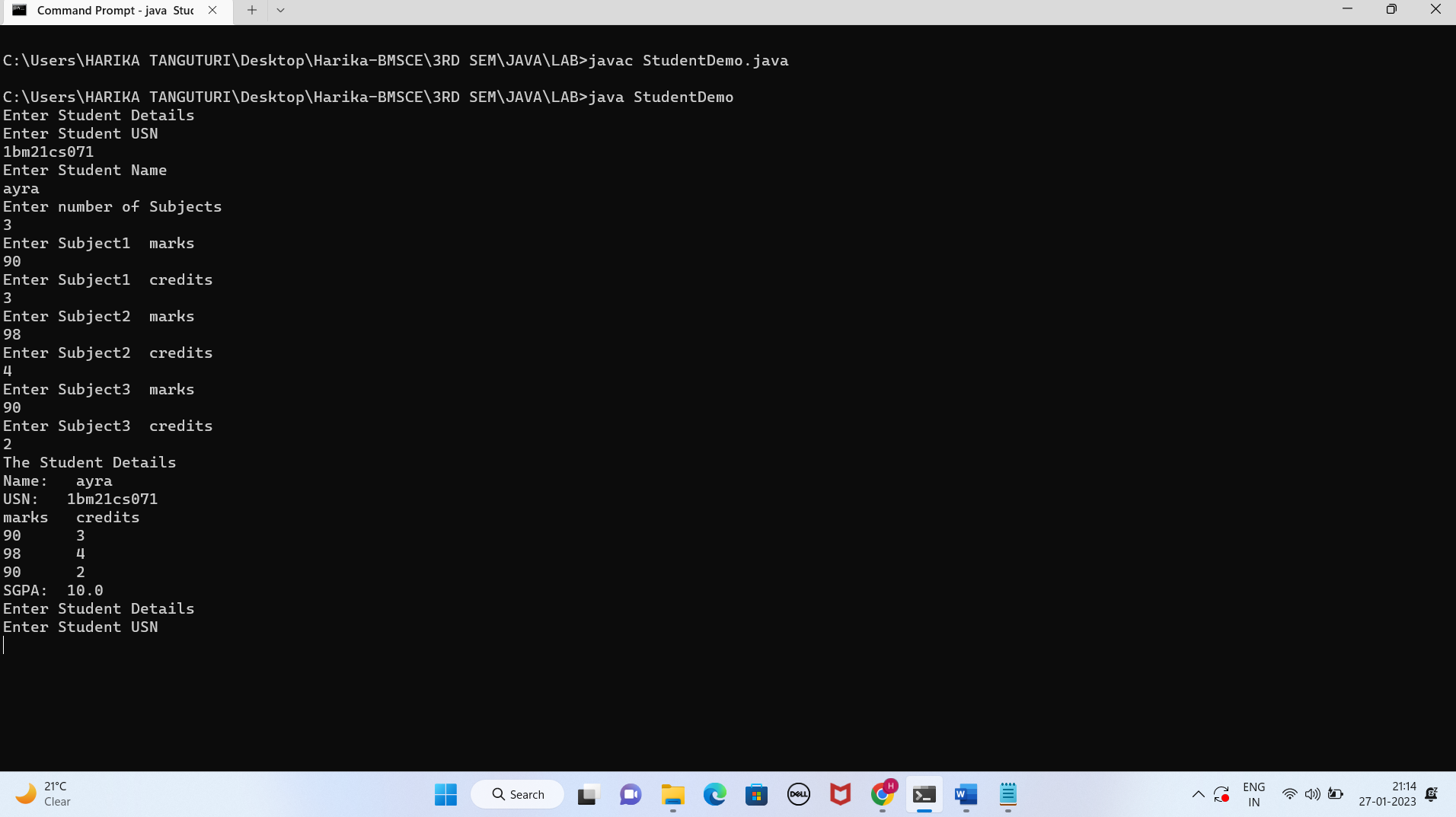
s2.calculate();

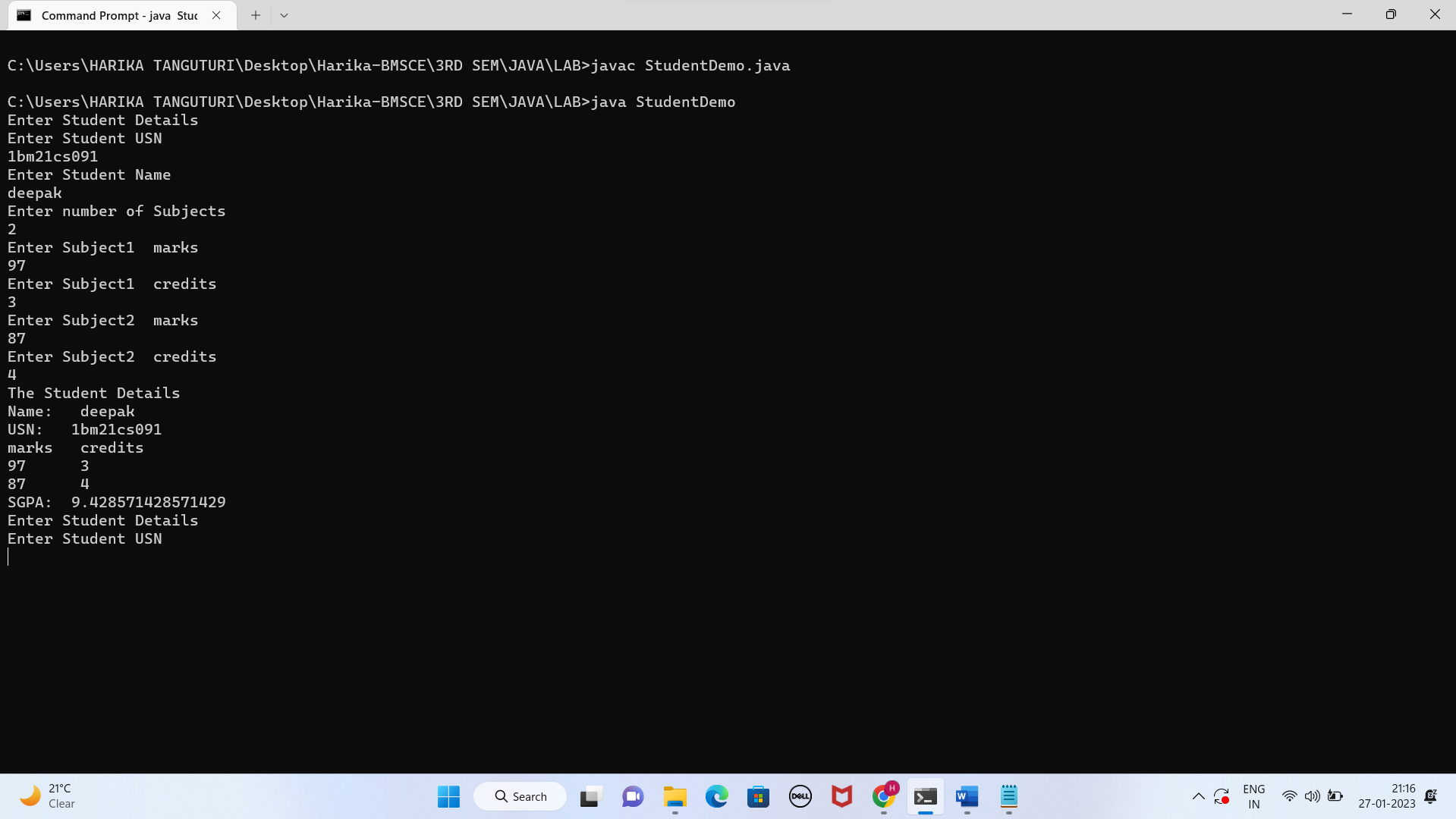
s2.display();

}

}

**OUTPUT:**

****

****

**LAB PROGRAM 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.\*;

class Book

{

String title;

String author;

double price;

int numPages;

Book()

{

title="Default";

author="Default";

price=0.00;

numPages=0;

}

void SetTitle(String t)

{

title=t;

}

void SetAuthor(String a)

{

author=a;

}

void SetPrice(double p)

{

price=p;

}

void SetPages(int np)

{

numPages=np;

}

public String toString()

{

return title+"\t\t"+author+"\t\t"+price+"\t\t"+numPages;

}

}

class BookDetails{

public static void main(String args[])

{

String t;

String a;

double p;

int np,n;

Scanner sc=new Scanner(System.in);

System.out.println("Enter the number of Books");

n=sc.nextInt();

Book b[]=new Book[n];

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

{

System.out.println("Enter the title of Book");

t=sc.next();

System.out.println("Enter the author of Book");

a=sc.next();

System.out.println("Enter the price of Book");

p=sc.nextDouble();

System.out.println("Enter the number of pages of Book");

np=sc.nextInt();

b[i]=new Book();

b[i].SetTitle(t);

b[i].SetAuthor(a);

b[i].SetPrice(p);

b[i].SetPages(np);

}

System.out.println("Title"+"\t"+"Author"+"\t"+"Price"+"\t"+"Pages");

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

{

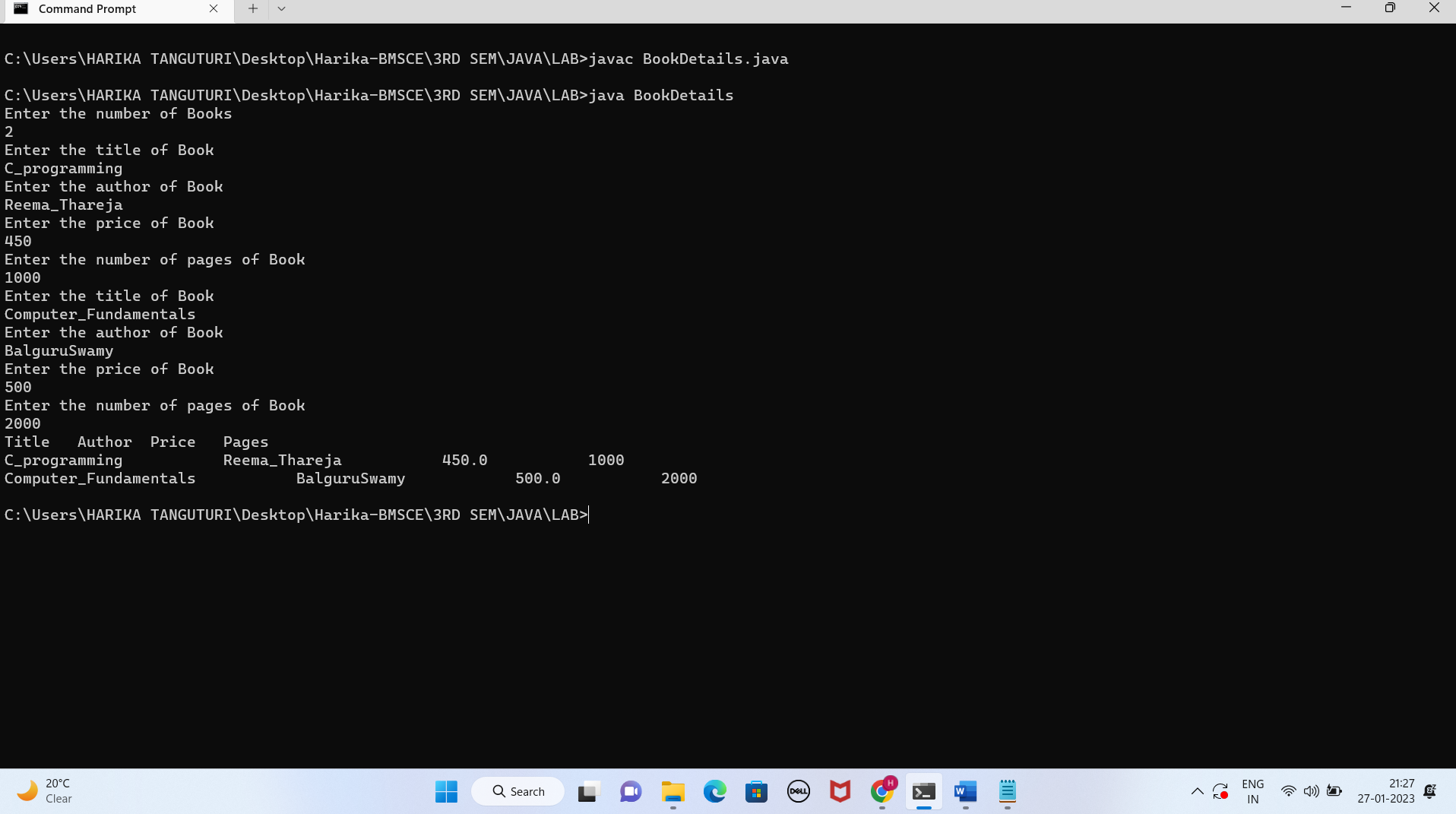
System.out.println(b[i]);

}

}

}

**OUTPUT:**

****

**LAB PROGRAM 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.**

import java.util.\*;

abstract class shape

{

int a,b;

abstract void printArea();

}

class rectangle extends shape

{

void getdata(int x,int y)

{

a=x;

b=y;

}

void printArea()

{

double x=a\*b;

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

}

}

class triangle extends shape

{

void getdata(int x,int y)

{

a=x;

b=y;

}

void printArea()

{

double area=0.5\*a\*b;

System.out.println("Area of Triangle is "+area);

}

}

class circle extends shape

{

void getdata(int x)

{

a=x;

}

void printArea()

{

double area=3.14\*a\*a;

System.out.println("Area of circle is "+area);

}

}

class abstr{

public static void main(String args[])

{

Scanner scan=new Scanner(System.in);

int choice;

rectangle r=new rectangle();

triangle t=new triangle();

circle c=new circle();

System.out.println("1.Rectangle 2.Triangle 3.Circle");

System.out.println("Enter your choice");

choice=scan.nextInt();

switch(choice)

{

case 1:System.out.println("Enter value of length and breadth");

int l=scan.nextInt();

int b=scan.nextInt();

r.getdata(l,b);

r.printArea();

break;

case 2:System.out.println("Enter value of base and height");

int b1=scan.nextInt();

int h=scan.nextInt();

t.getdata(b1,h);

t.printArea();

break;

case 3:System.out.println("Enter value of radius");

int r1=scan.nextInt();

c.getdata(r1);

c.printArea();

break;

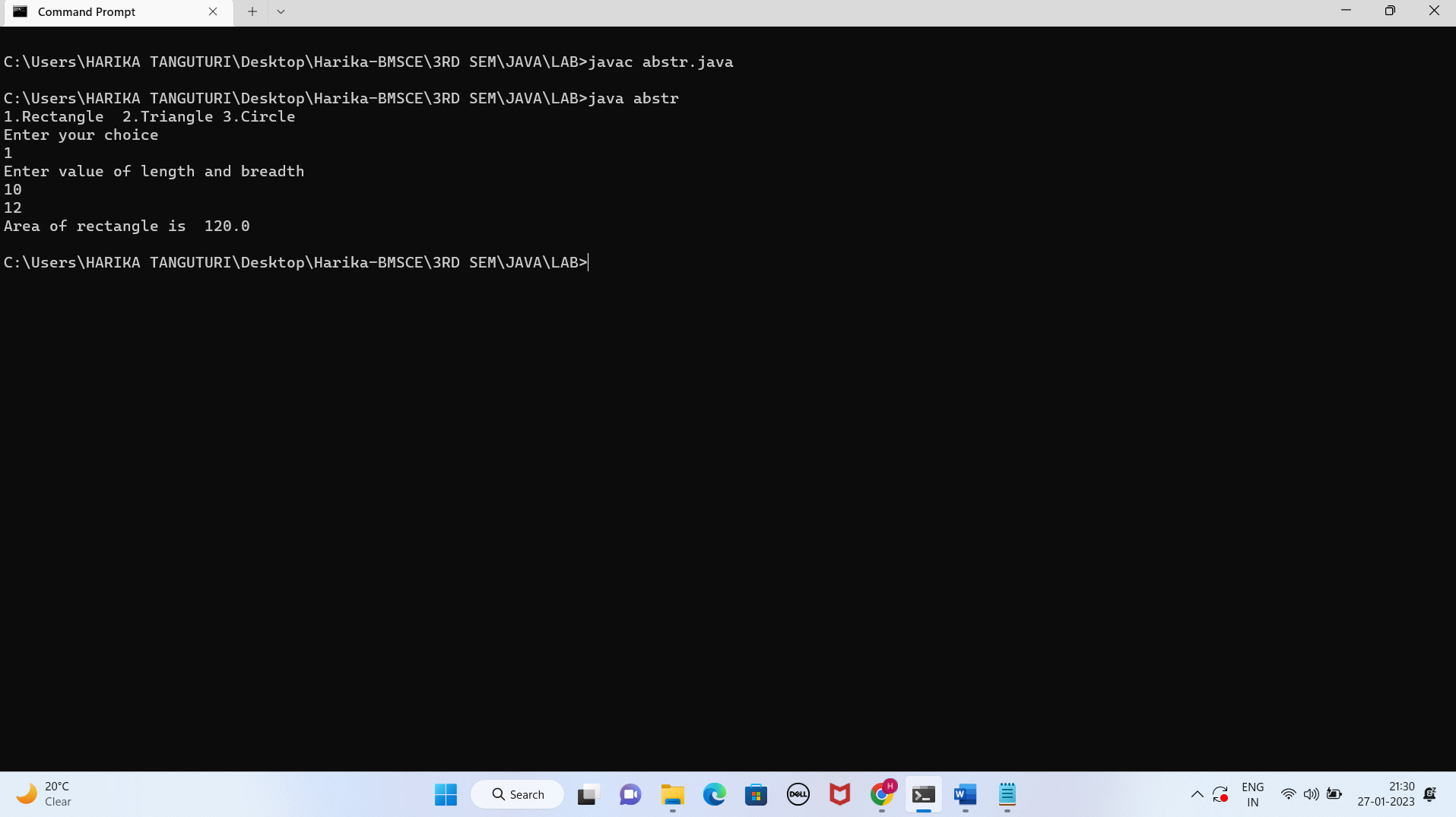
default:System.out.println("Invalid choice!!!");

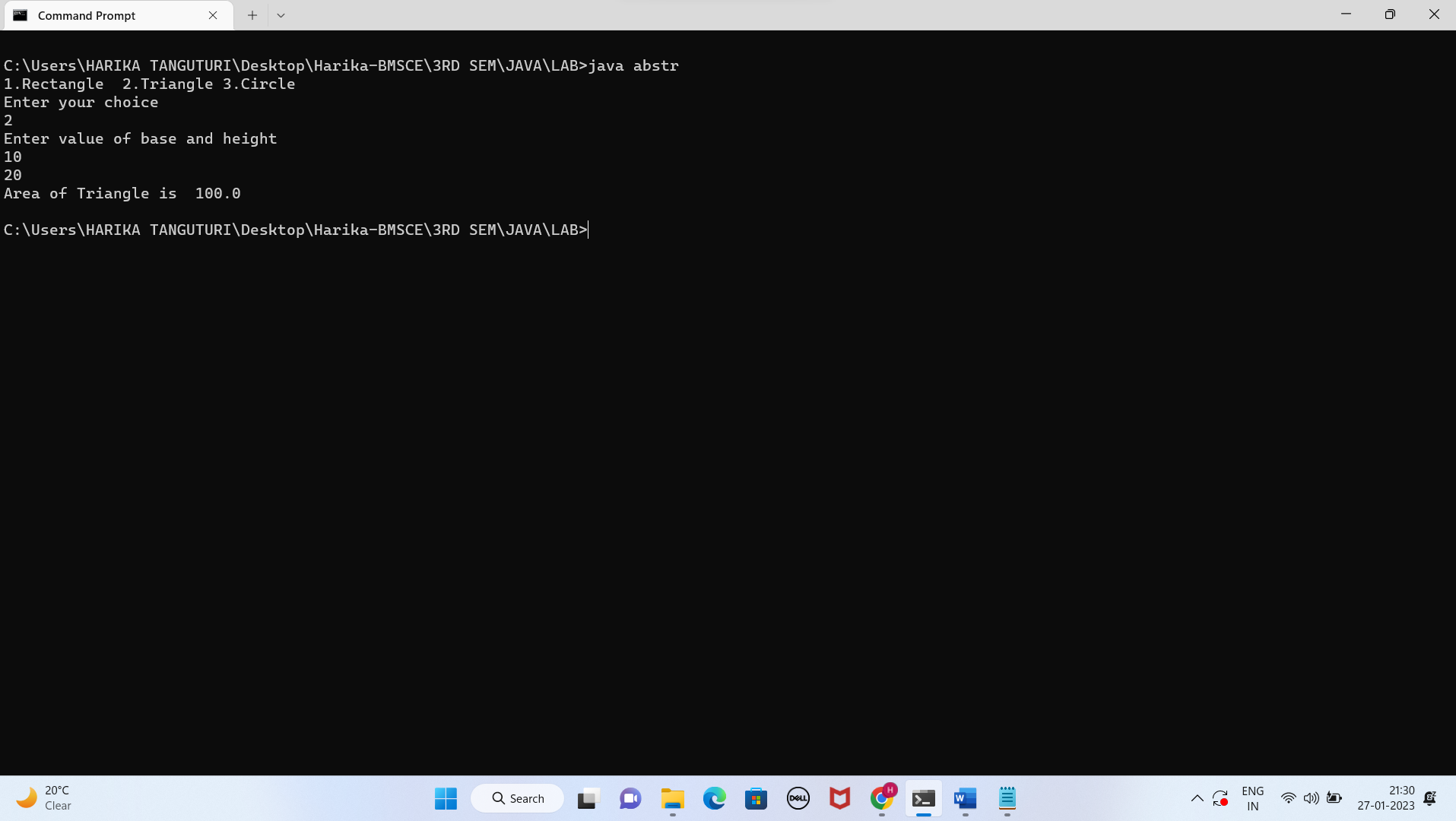
}

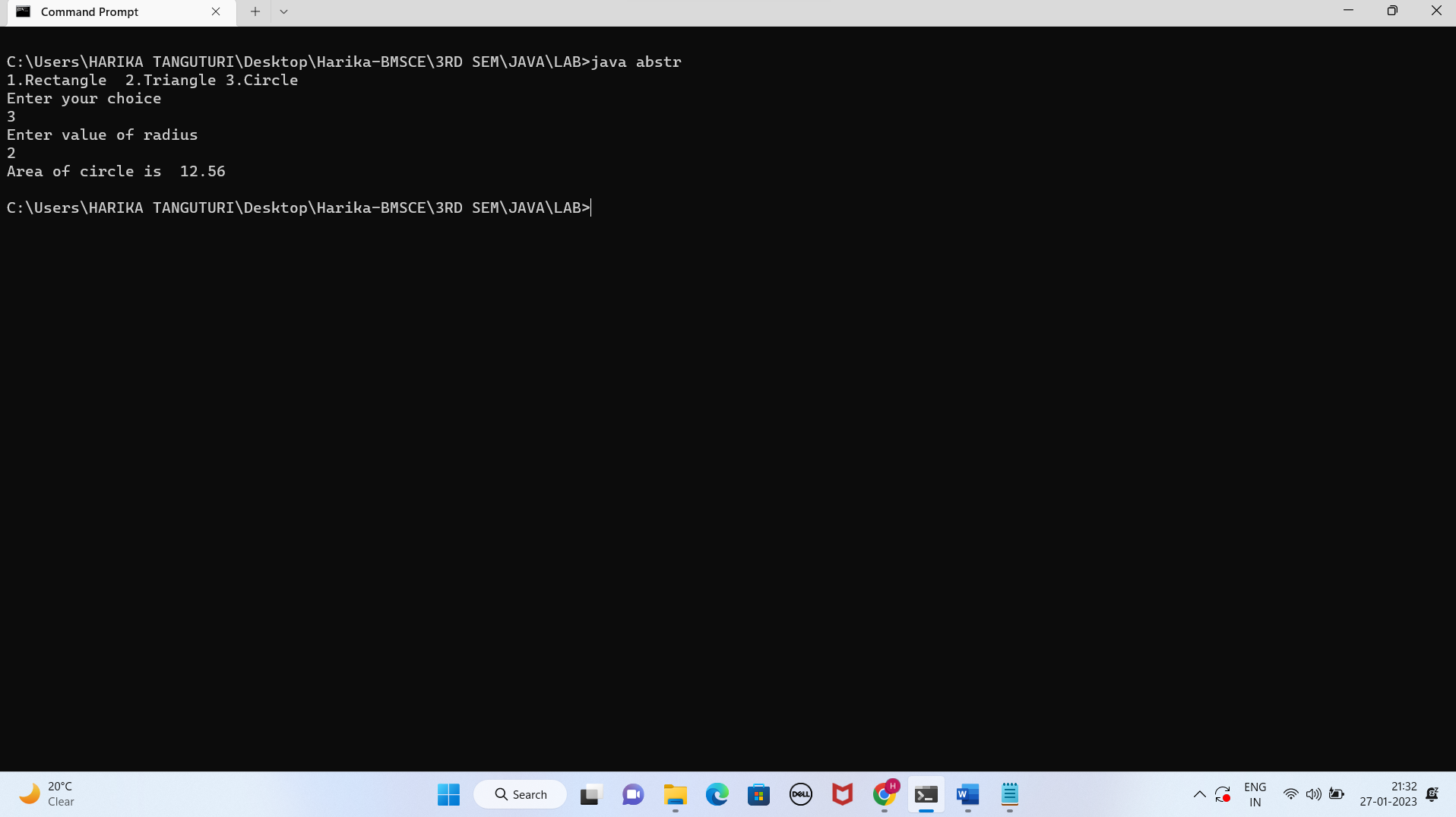
}

}

**OUTPUT:**

****

****

****

**LAB PROGRAM 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;

class Account

{

String customer\_name;

long acc\_no;

float bal;

Scanner s = new Scanner(System.in);

public void input()

{

System.out.print("\nEnter the Customer Name: ");

customer\_name = s.nextLine();

System.out.print("\nEnter the Account Number: ");

acc\_no = s.nextLong();

System.out.print("\nEnter the Starting Amount (Minimum Amount = 5000): ");

bal = s.nextFloat();

if(bal<5000f)

{

System.out.println("\nAccount Balance cannot be less than 5000.0 \n");

System.exit(0);

}

}

public void display()

{

System.out.println("\nCustomer Name: "+customer\_name);

System.out.println("Account Number: "+acc\_no);

System.out.println("Amount: "+bal);

}

}

class Savings extends Account

{

Scanner s = new Scanner(System.in);

float deposit,withdraw,interest;

public void deposit()

{

System.out.print("\nEnter the amount to be deposited: ");

deposit = s.nextFloat();

bal+=deposit;

System.out.println("\nBalance: "+bal);

}

public void withdraw()

{

System.out.print("\nEnter the amount to be withdrawn: ");

withdraw = s.nextFloat();

if(bal<5000)

{

System.out.println("\nInsufficient Balance");

}

else

{

bal-=withdraw;

System.out.println("\nAmount Withdrawn: "+withdraw+"\nBalance: "+bal);

}

}

public void check\_Bal()

{

if(bal<5000)

{

System.out.println("\nInsufficient Balance!!\nBalance: "+bal);

}

else

{

System.out.println("\nBalance: "+bal);

}

}

public void interest()

{

interest=(bal\*6)/100;

bal+=interest;

System.out.println("\nInterest Credited: "+interest+"\nBalance :"+bal) ;

}

}

class Current extends Account

{

float deposit, withdraw, penalty;

public void deposit()

{

System.out.print("\nEnter Amount to be deposited: ");

deposit = s.nextFloat();

bal += deposit;

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

}

public void check\_Bal()

{

if (bal < 5000)

{

penalty = (0.1f \* bal);

System.out.println("\nInitial Account Balance: "+bal);

bal = bal-penalty;

System.out.println("\nLow balance!\nPenalty Amount: " + penalty + "\nAccount balance: " + bal);

}

else

{

System.out.println("\n Balance: " + bal);

}

}

public boolean check\_Bal\_part\_2()

{

if (bal < 5000)

{

penalty = (0.1f \* bal);

System.out.println("\nInitial Account Balance: "+bal);

bal = bal-penalty;

System.out.println("\nLow Balance!\nPenalty Amount: " + penalty + "\nAccount balance: " + bal);

return false;

}

return true;

}

public void withdraw()

{

System.out.print("\nEnter Amount to withdraw: ");

withdraw = s.nextFloat();

if(check\_Bal\_part\_2())

{

bal-=withdraw;

System.out.println("\nAmount Withdrawn: "+withdraw+"\nBalance: "+bal);

}

}

public void chequebook()

{

System.out.println("\nCheque Book has been Issued!");

}

}

public class Bank

{

public static void main(String[] args)

{

Scanner s = new Scanner(System.in);

String ch;

int n;

Current c = new Current();

Savings sa = new Savings();

System.out.print("\nEnter the Account Type (S for Savings , C for Current) : ");

ch = s.next();

switch(ch.toLowerCase())

{

case "s" : sa.input();

do

{

System.out.print("\n1. Deposit \n2. Withdrawal \n3. Check Balance \n4. Check Interest"

+"\n5. Show Account Details \n6. Exit Transaction\n\nEnter your choice: ");

n = s.nextInt();

switch(n)

{

case 1 : sa.deposit();

break;

case 2 : sa.withdraw();

break;

case 3 : sa.check\_Bal();

break;

case 4 : sa.interest();

break;

case 5 : sa.display();

break;

case 6 : System.out.println("\nExiting Transaction!");

System.exit(0);

break;

default : System.out.println("\nInvalid Operation");

}

}while(true);

case "c" : c.input();

do {

System.out.print("\n1. Deposit \n2. Withdrawal \n3. Check Balance \n4. Issue Cheque Book"

+ "\n5. Show Account Details \n6. Exit Transaction\n\nEnter your choice: ");

n = s.nextInt();

switch (n) {

case 1:

c.deposit();

break;

case 2:

c.withdraw();

break;

case 3:

c.check\_Bal();

break;

case 4:

c.chequebook();

break;

case 5:

c.display();

break;

case 6:

System.out.println("\nExiting Transaction!");

System.exit(0);

break;

default:

System.out.println("\nInvalid Operation");

}

}while(true);

default : System.out.println("\nInvalid Choice");

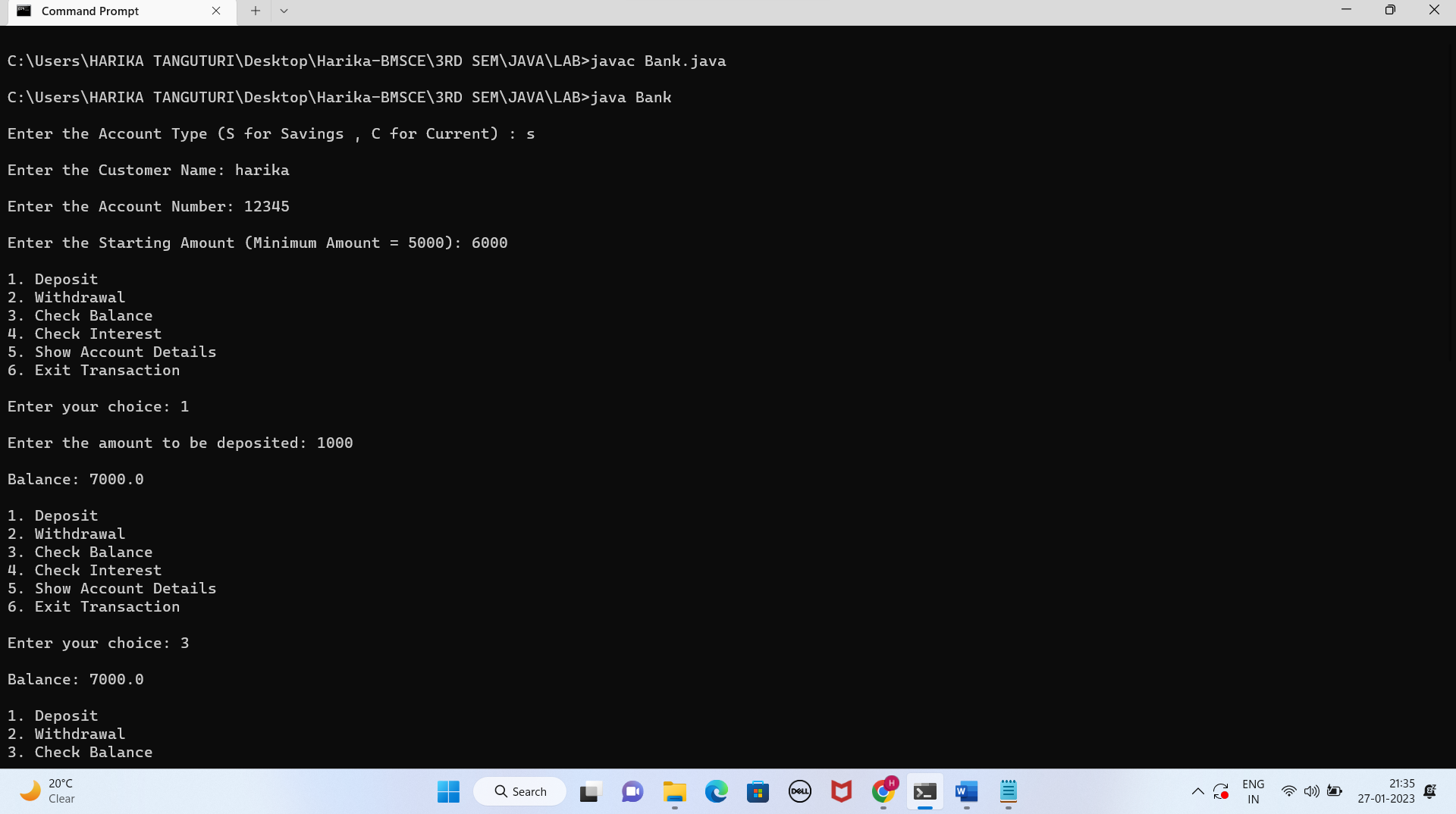
break;

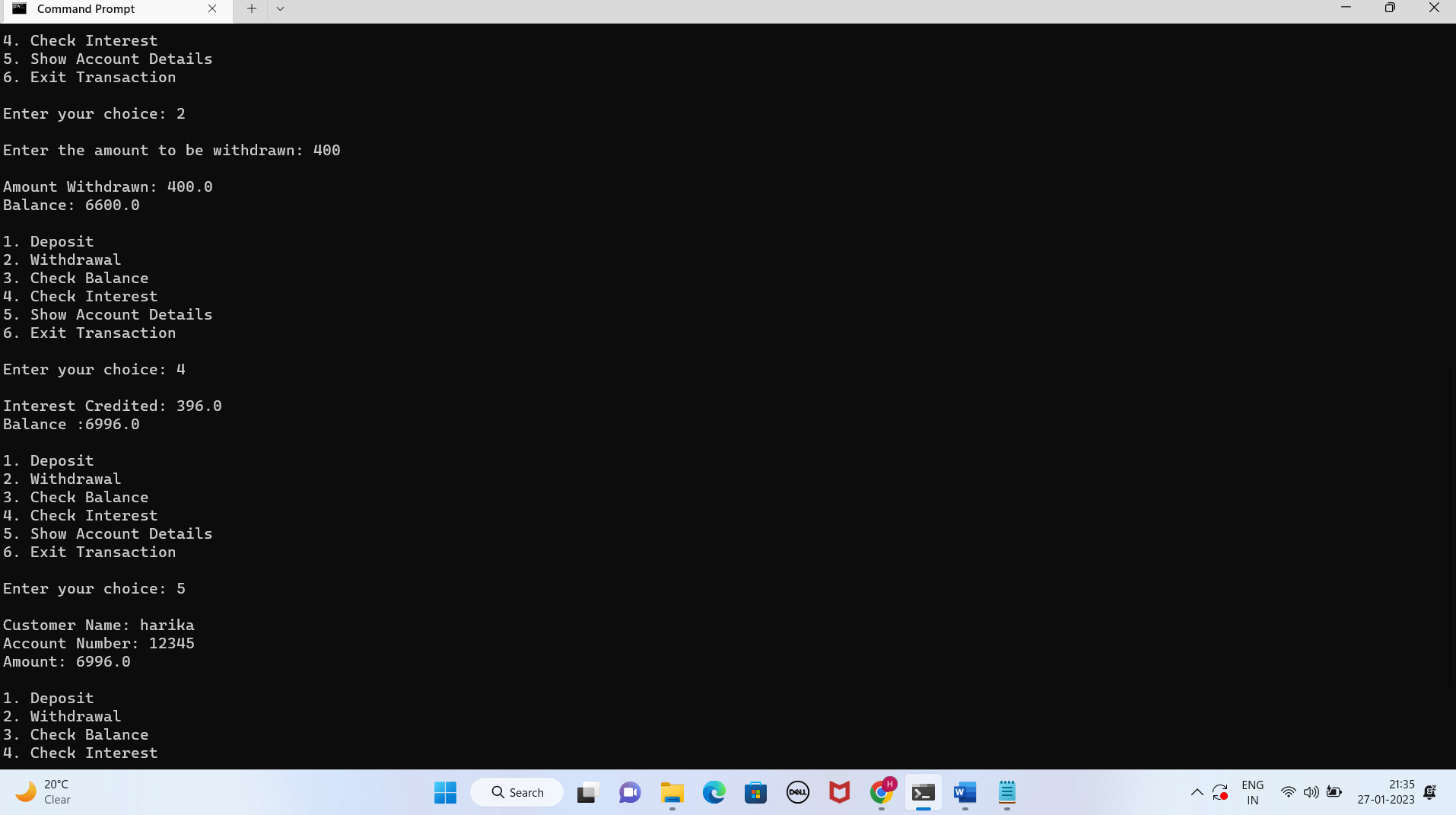
}

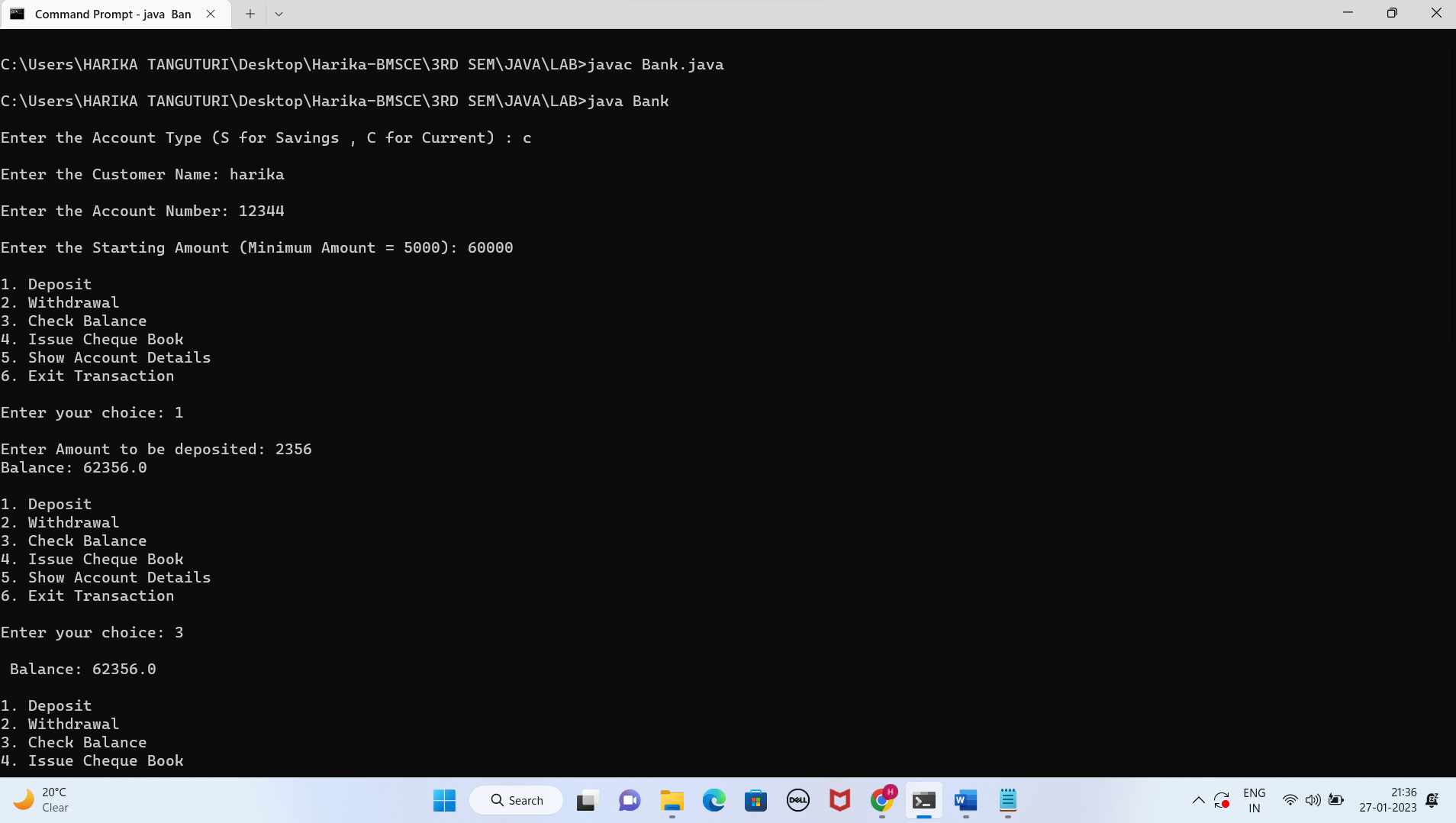
}

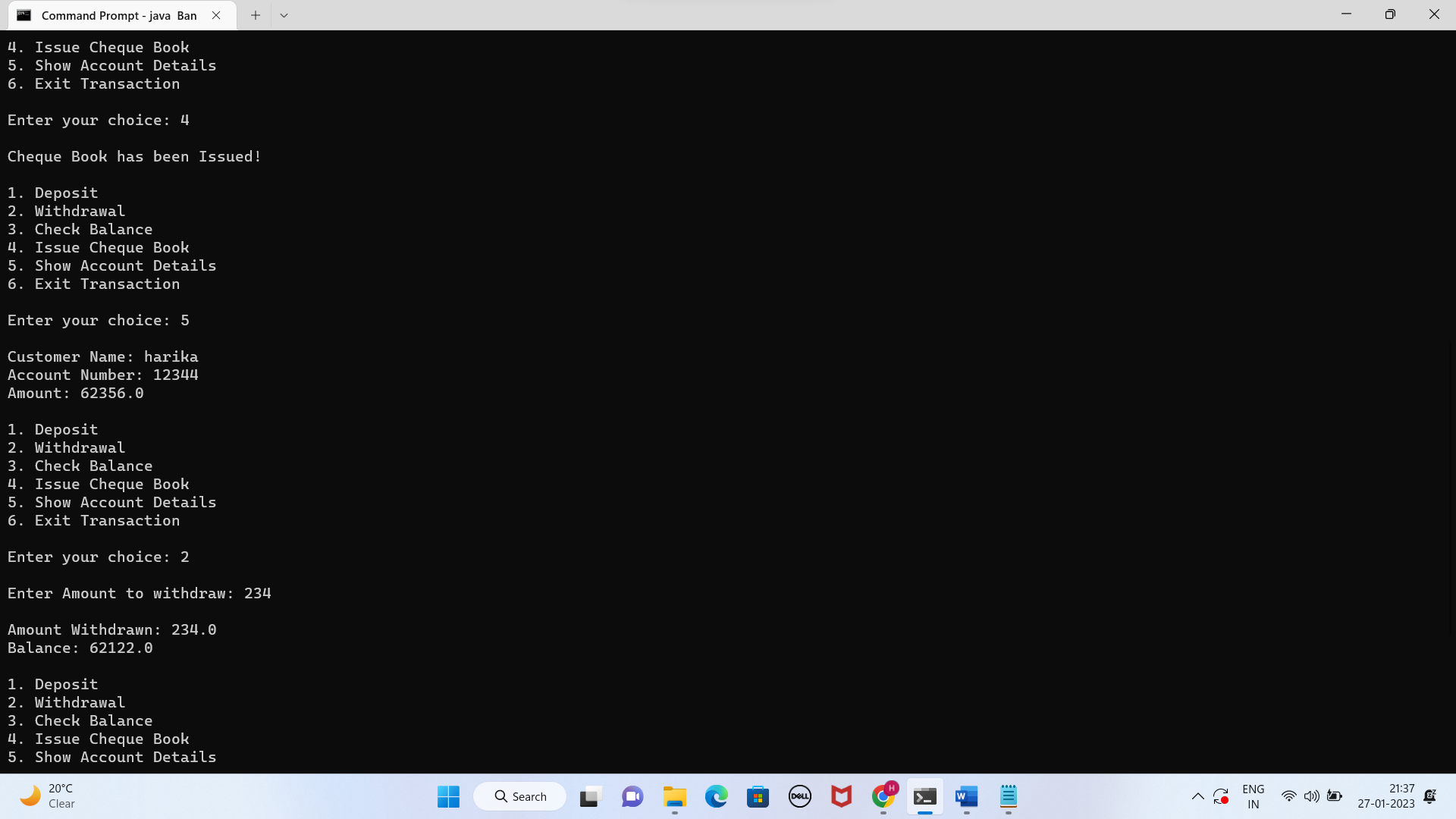
}

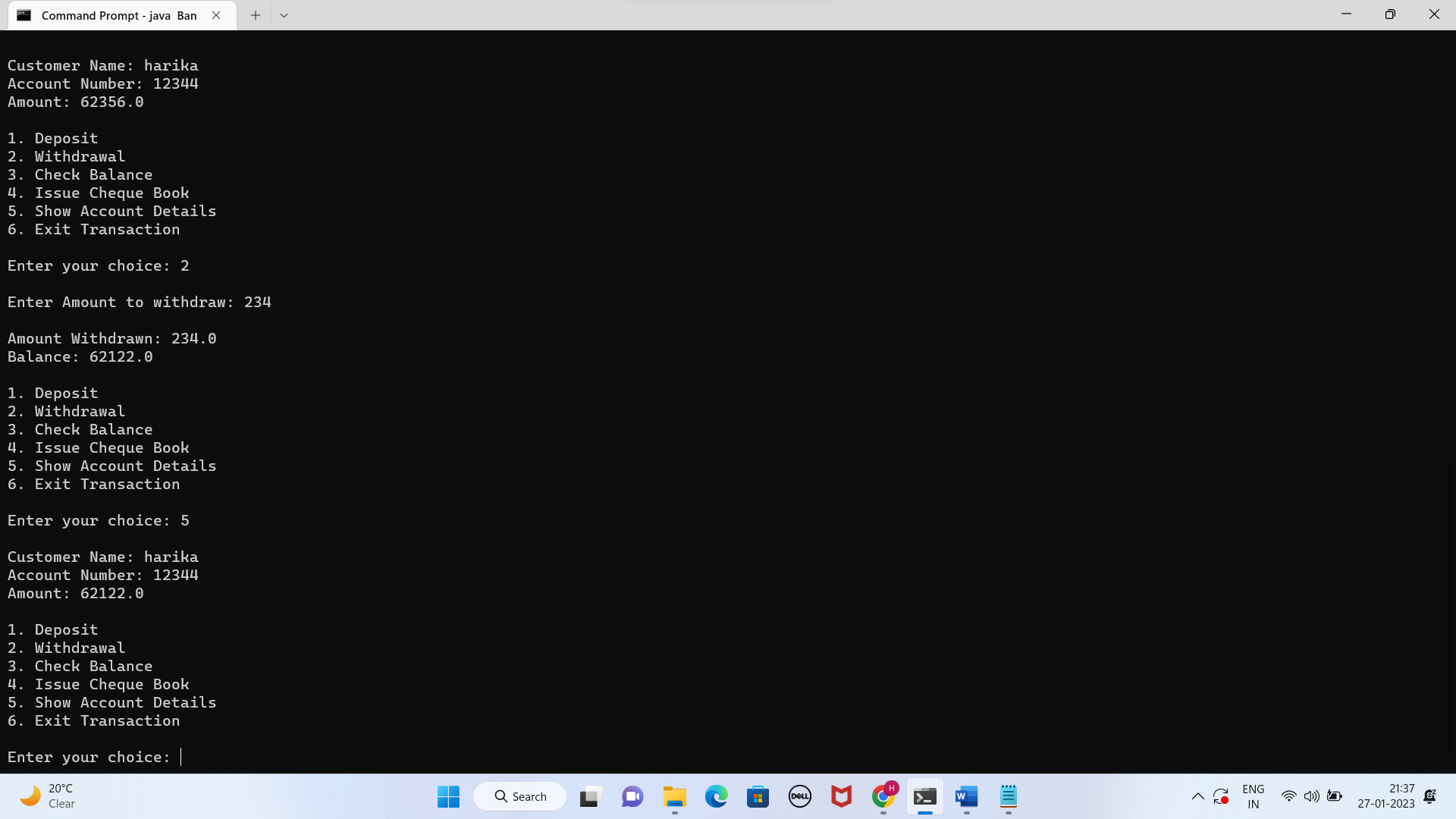
**OUTPUT:**

****

****

****

****

****

**LAB PROGRAM 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 WrongAge extends Exception{

public String detail;

WrongAge(String a){

detail=a;

}

public String toString(){

return "WrongAge["+detail+"]";

}

}

class Father{

int father\_age;

Father(int x)

{

father\_age=x;

}

}

class Son extends Father{

int son\_age;

Son(int x,int y)

{

super(x);

son\_age=y;

try{

if(son\_age<=0 || father\_age<=0)

{

throw new WrongAge("Son's age or Father's age is less than or equal to zero");

}

if(father\_age<=son\_age)

{

throw new WrongAge("Son's age is greater than or equal to Father's age");

}

else

{

System.out.println("Entered Age is Valid!!!");

}

}

catch(WrongAge e){

System.out.println("caught"+e);

}

}

}

class Age{

public static void main(String args[]){

Scanner sc=new Scanner(System.in);

int father\_age,son\_age;

System.out.println("Enter father age");

father\_age=sc.nextInt();

System.out.println("Enter son's age");

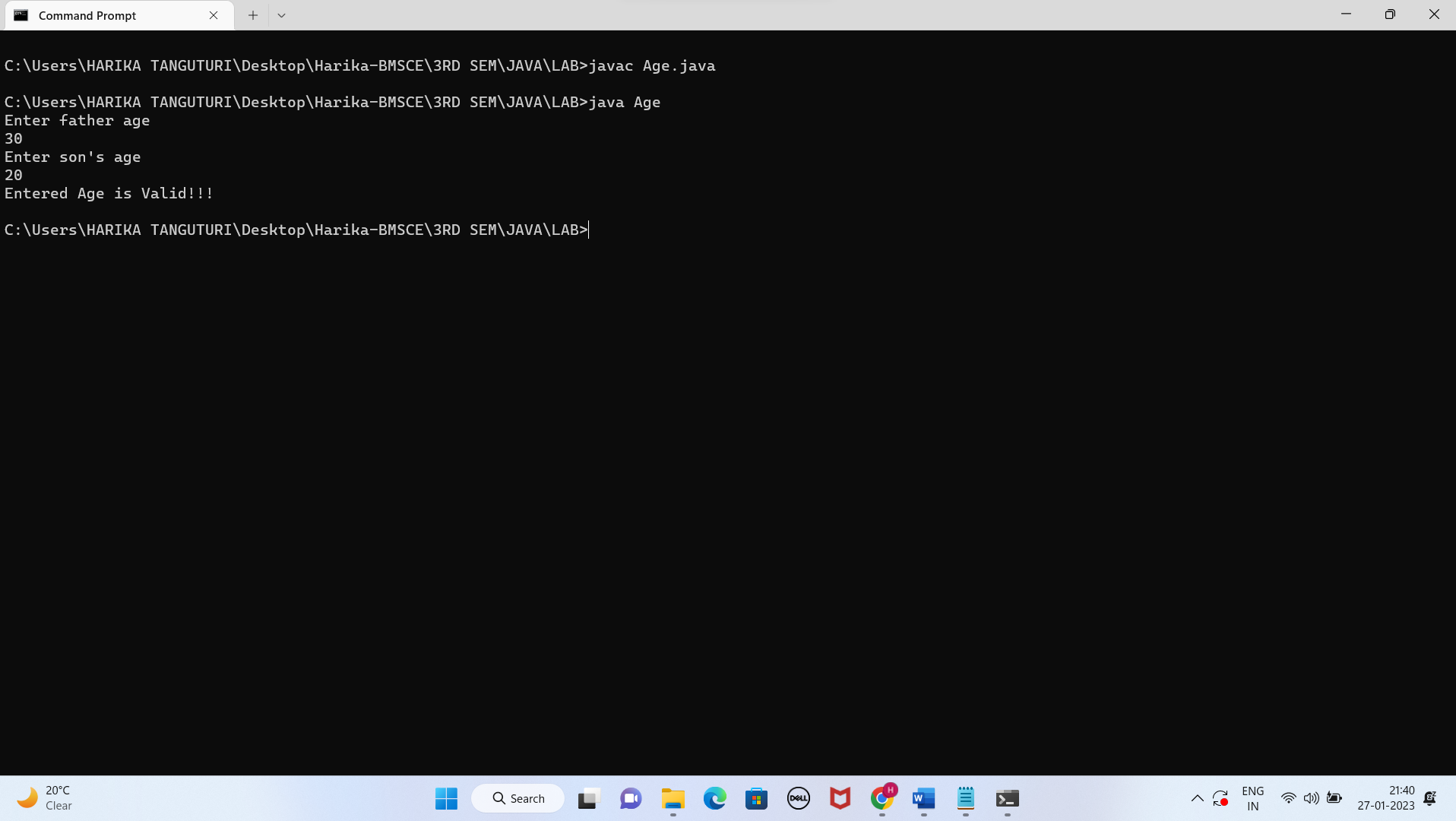
son\_age=sc.nextInt();

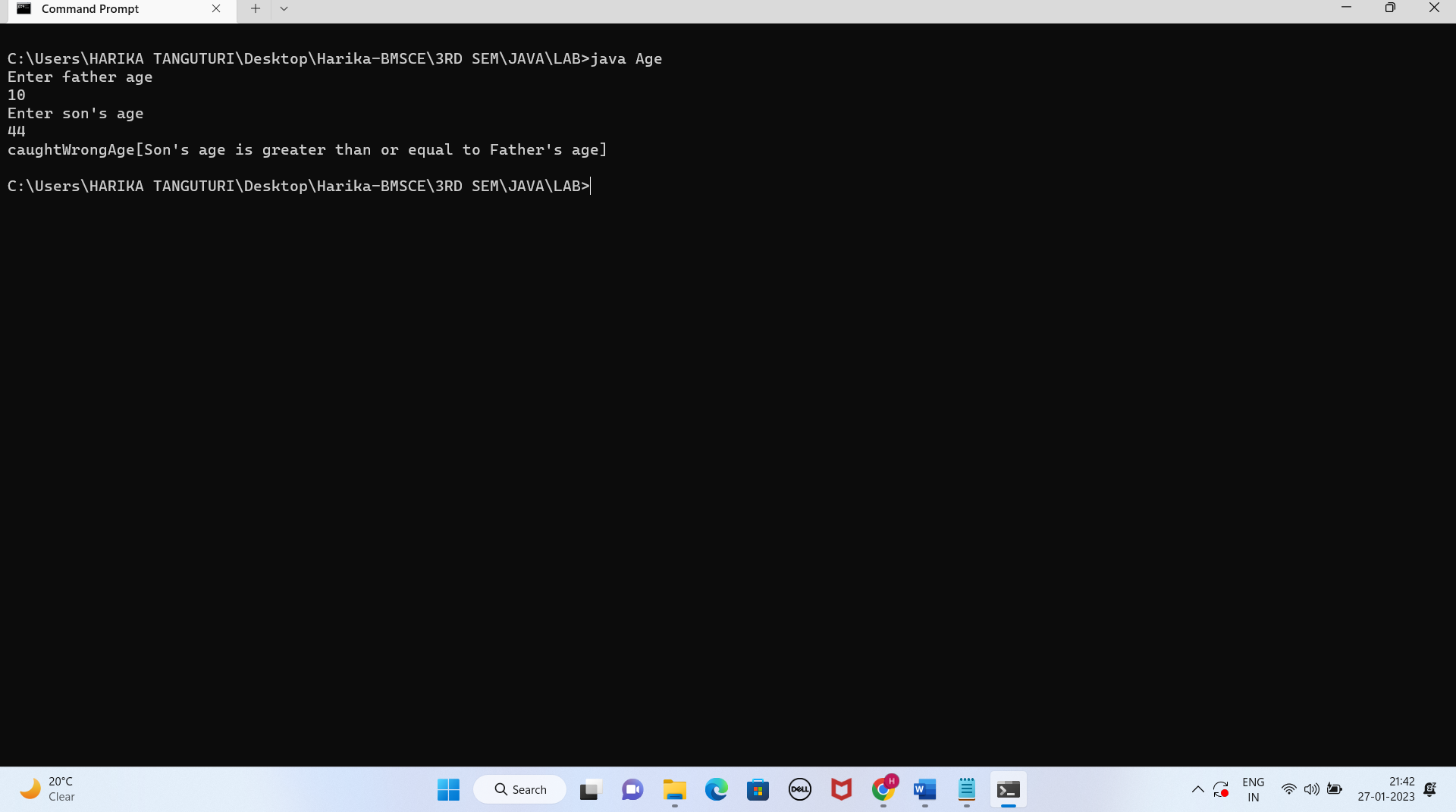
Son s=new Son(father\_age,son\_age);

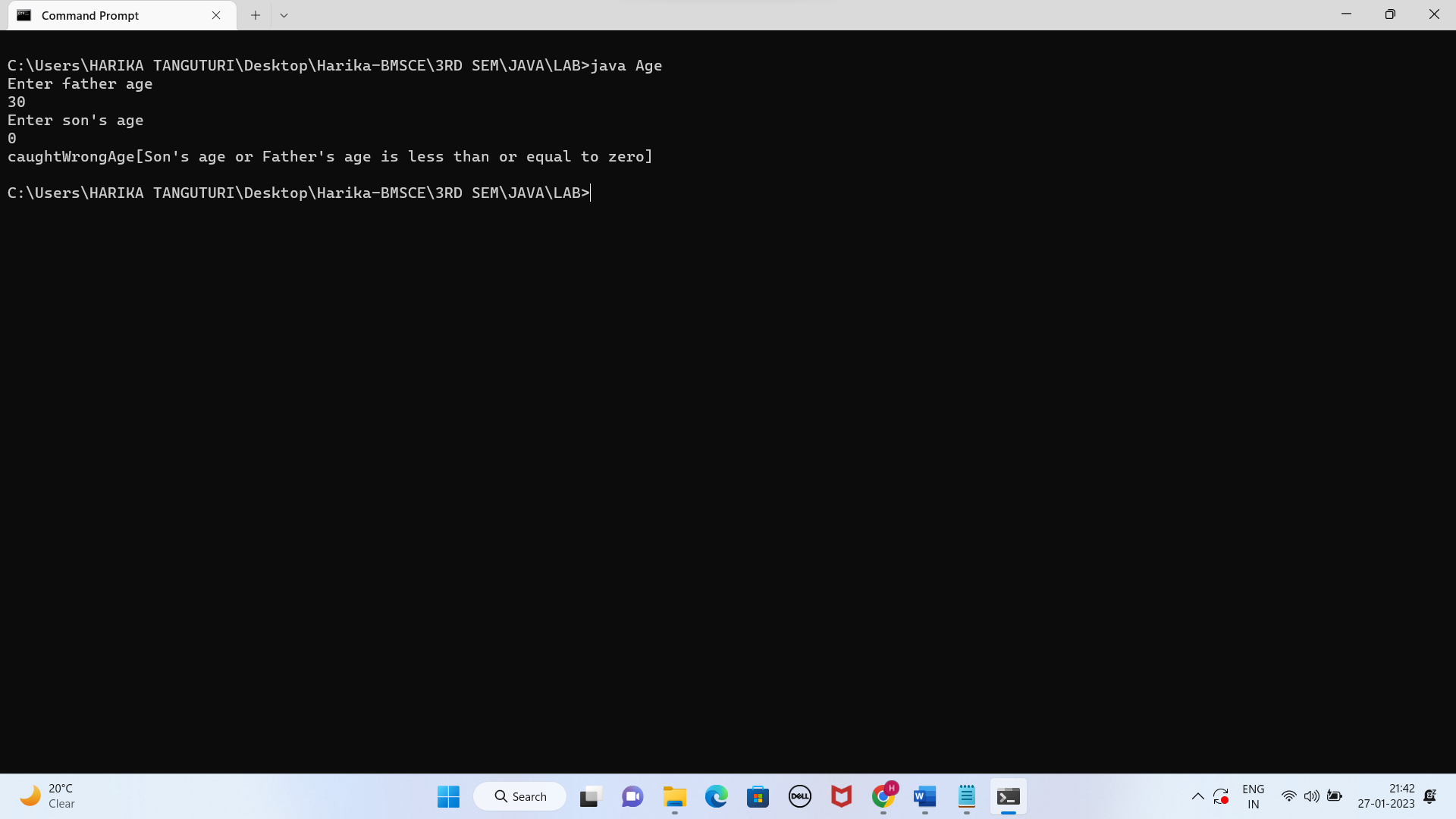
}

}

**OUTPUT:**

****

****

****

**LAB PROGRAM 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 Thread1

{

public static void main(String xx[])

{

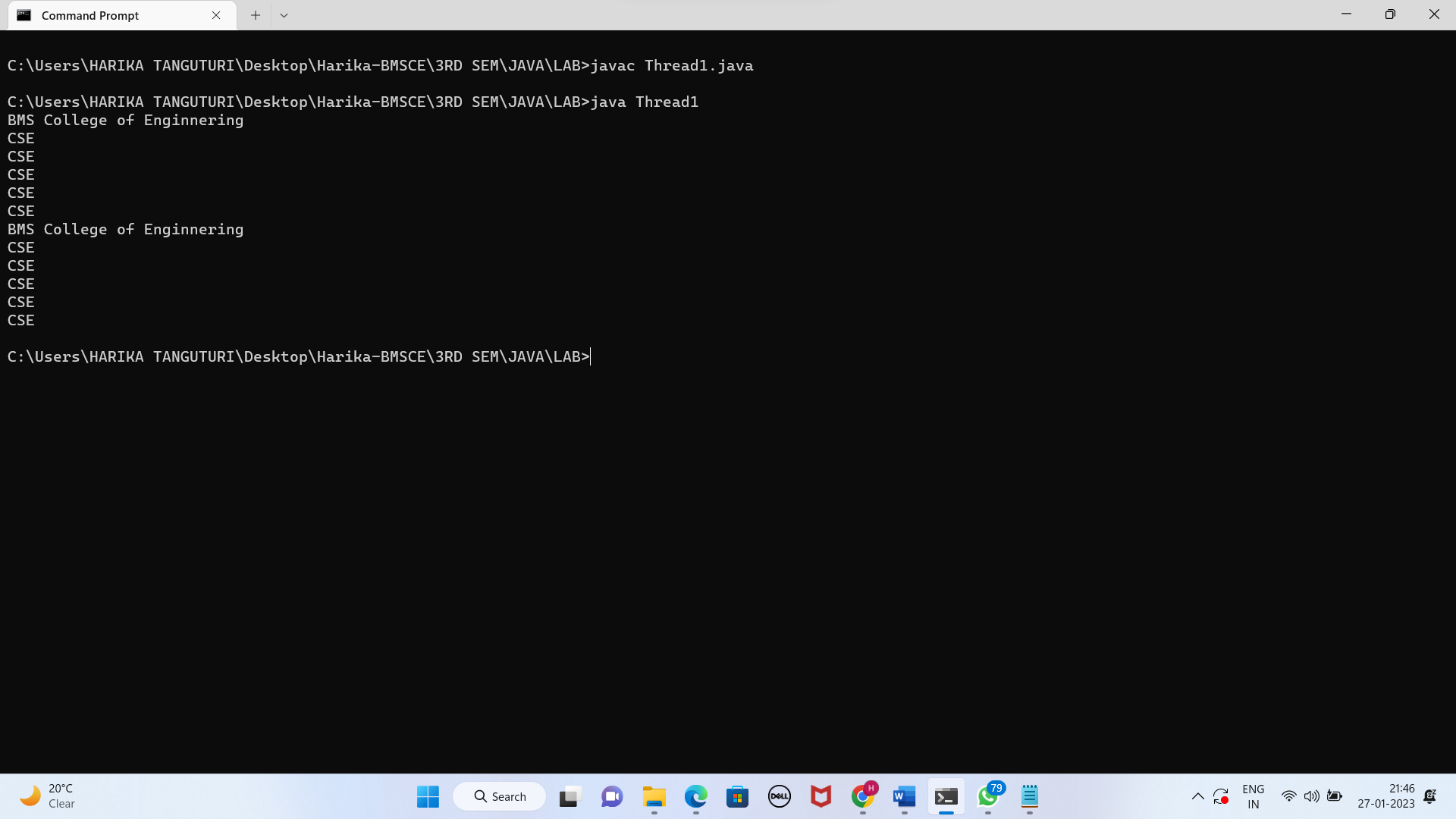
new Call("BMS College of Enginnering",10000,2);

new Call("CSE",2000,10);

}

}

**OUTPUT:**

****