LAB PROGRAMS

1.EVEN OR ODD

Aim: Write a program to check whether a number is odd or even.

Code:

import java.util.Scanner;

class oddeven {

public static void main(String[] args) {

Scanner reader=new Scanner(System.in);

System.out.print("Enter a number:");

int num=reader.nextInt();

if(num%2==0)

System.out.print("Even number");

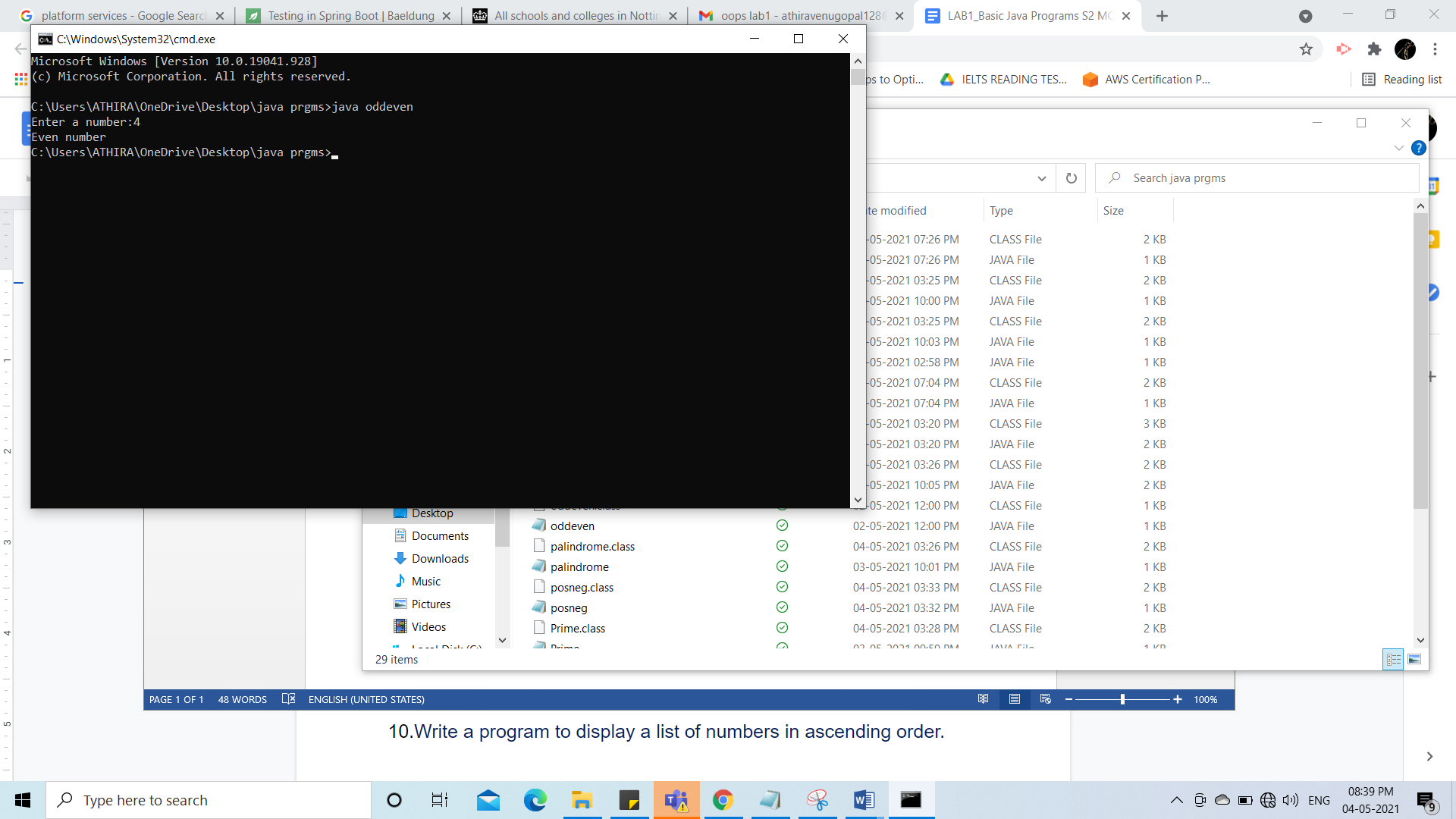
else

System.out.print(" Odd number");

}

}

Output



2.LARGEST OF 3 NUMBERS

Aim: Write a program to find the largest of three numbers.

Code:

import java.util.Scanner;

class biggest {

public static void main(String[] args) {

Scanner reader=new Scanner(System.in);

System.out.print("Enter first number:");

int num1=reader.nextInt();

System.out.print("Enter second number:");

int num2=reader.nextInt();

System.out.print("Enter third number:");

int num3=reader.nextInt();

if(num1>num2&&num1>num3)

System.out.print("The number "+num1+" is bigger");

else if(num2>num1&&num2>num3)

System.out.print("The number "+num2+" is bigger");

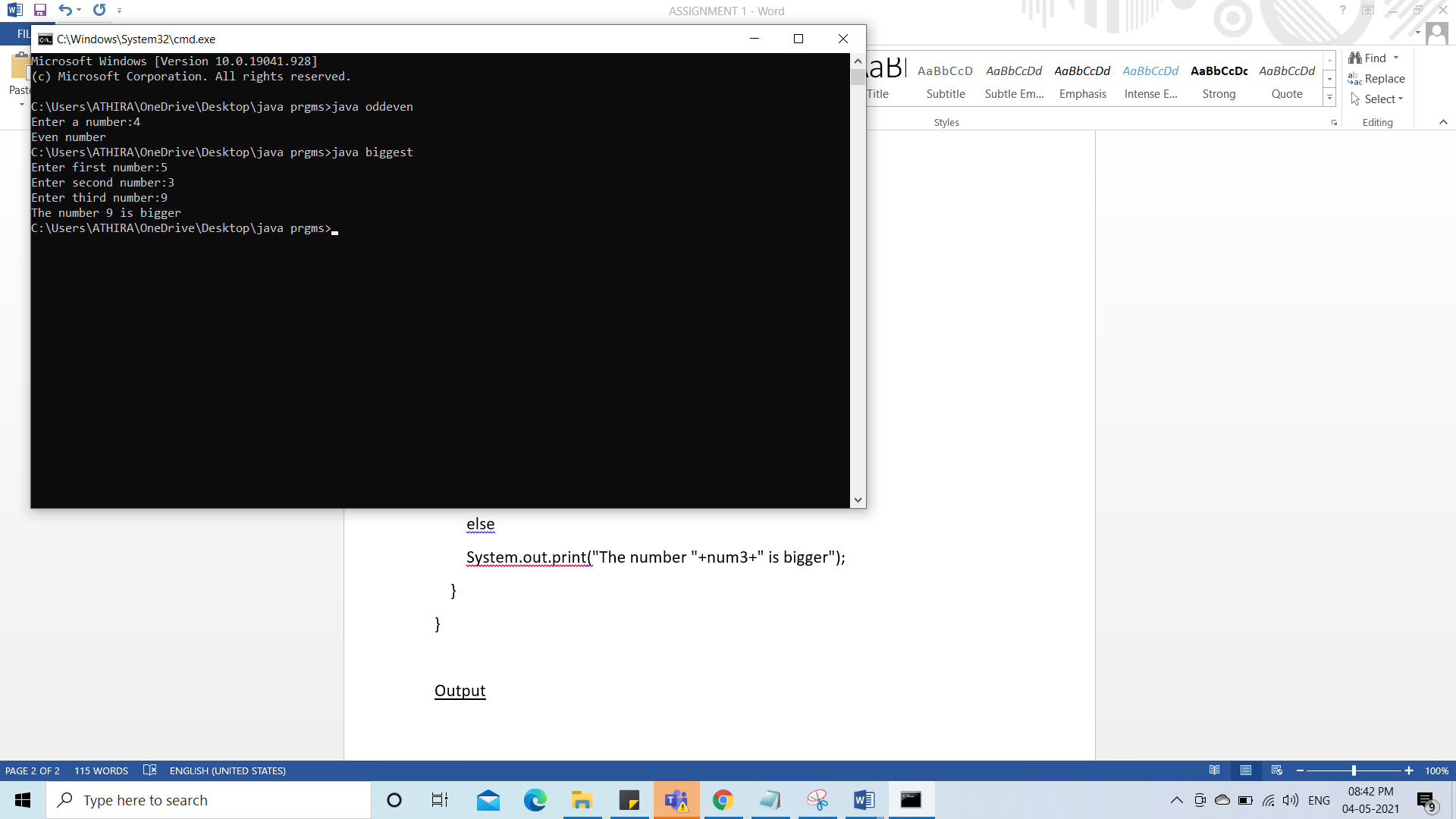
else

System.out.print("The number "+num3+" is bigger");

}

}

Output



3.POSITIVE OR NEGATIVE

Aim: Write a program to check whether a given number is positive or negative using  ternary operator. ( ?: )

Code:

import java.util.Scanner;

class posneg {

public static void main(String[] args) {

Scanner reader=new Scanner(System.in);

System.out.print("Enter a number:");

int num=reader.nextInt();

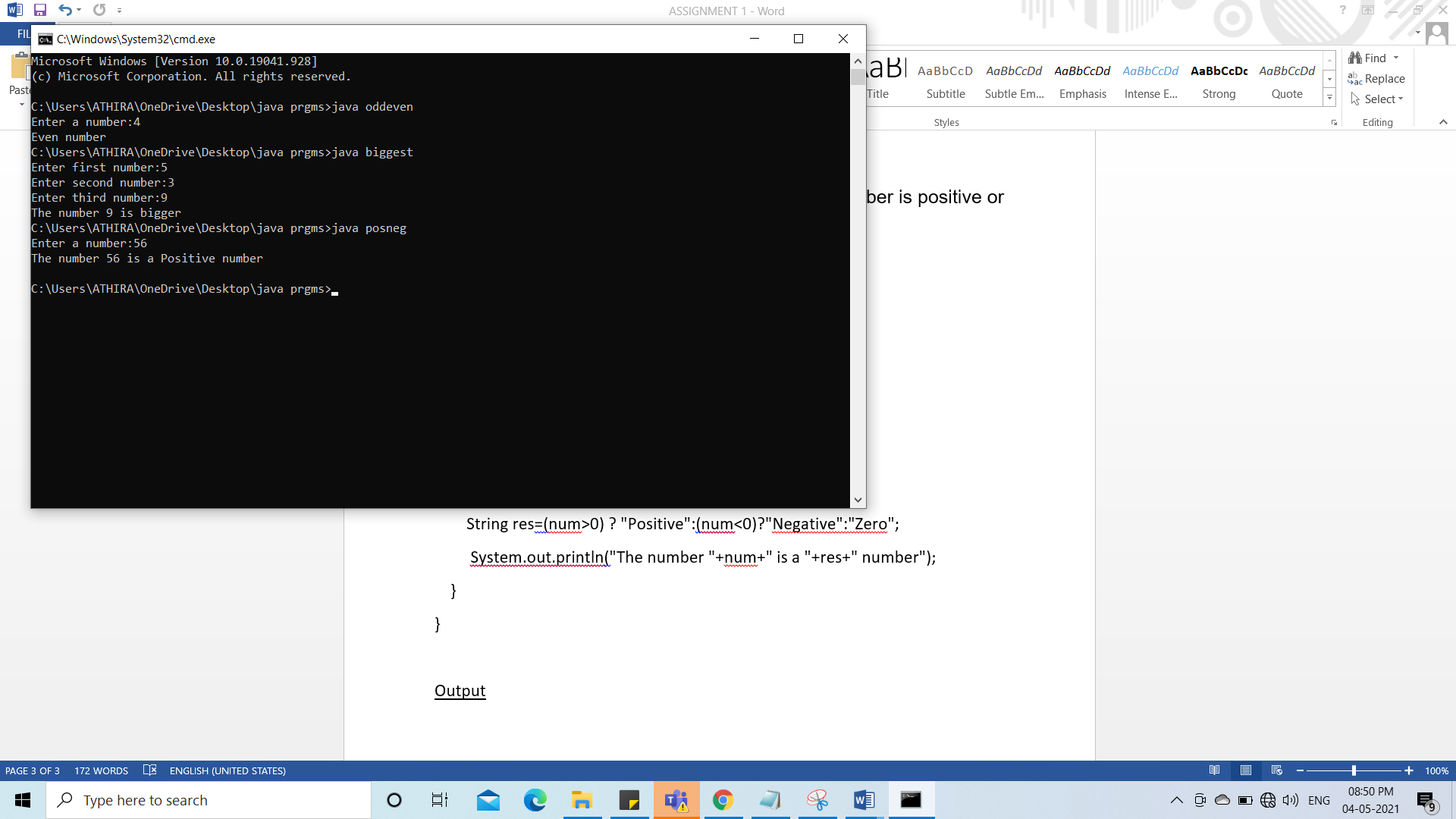
String res=(num>0) ? "Positive":(num<0)?"Negative":"Zero";

System.out.println("The number "+num+" is a "+res+" number");

}

}

Output



4.FACTORIAL

Aim: Write a program to find the factorial of a number.

Code:

import java.util.Scanner;

class factorial {

public static void main(String[] args) {

Scanner reader=new Scanner(System.in);

System.out.print("Enter a number:");

int num=reader.nextInt();

int i,fact=1;

for(i=1;i<=num;i++){

fact=fact\*i;

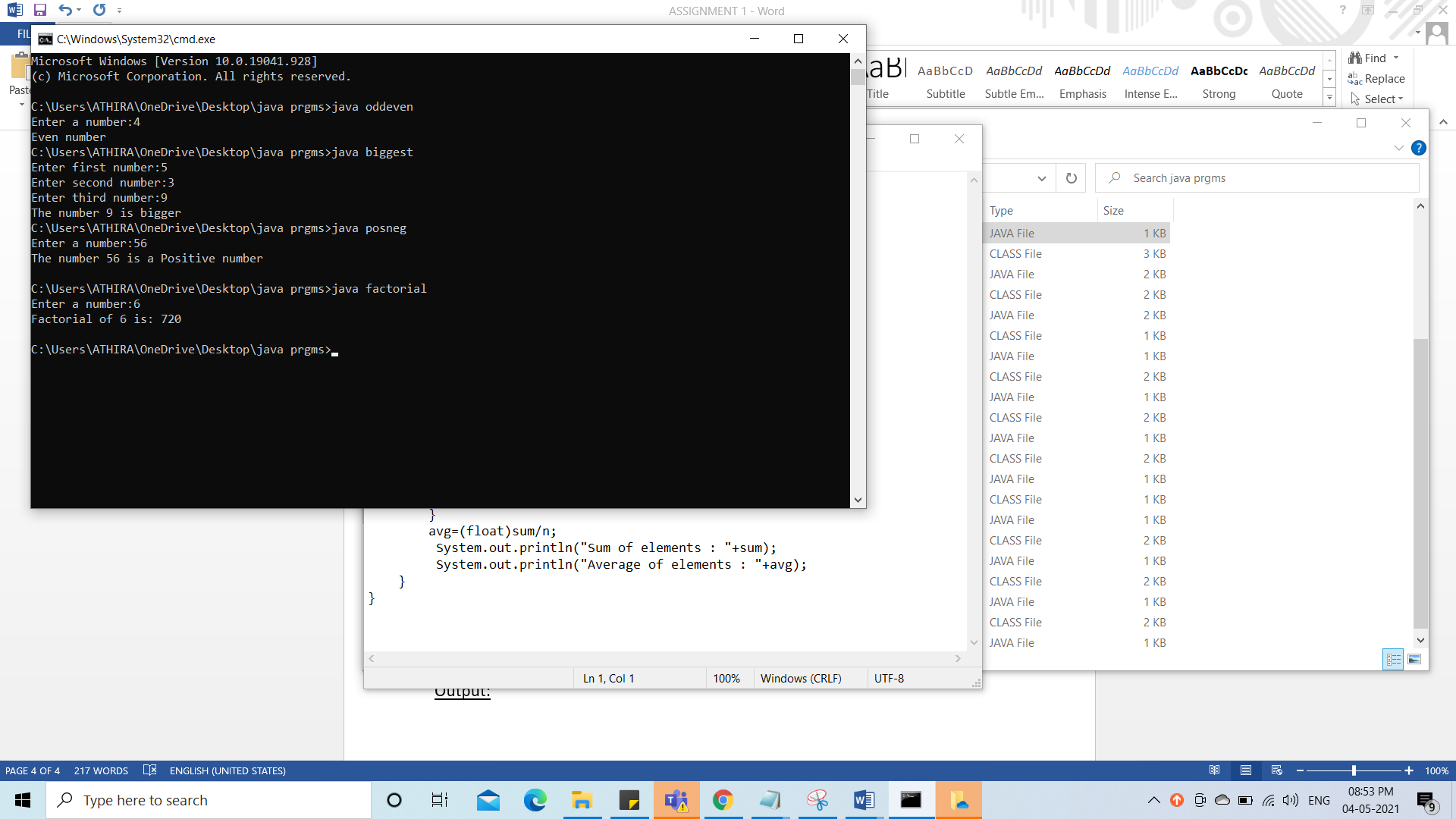
}

System.out.println("Factorial of "+num+" is: "+fact);

}

}

Output:



5.PALINDROME

Aim: Write a program to check whether a number is palindrome or not.

Code:

import java.util.Scanner;

class palindrome {

public static void main(String[] args) {

Scanner reader=new Scanner(System.in);

System.out.print("Enter a number:");

int num=reader.nextInt();

int temp=num;

int rev=0;

while(num!=0)

{

int a=num%10;

rev=rev\*10+a;

num=num/10;

}

System.out.println("Reversed number is "+rev);

if(temp==rev)

System.out.println("The number "+temp+" is palindrome");

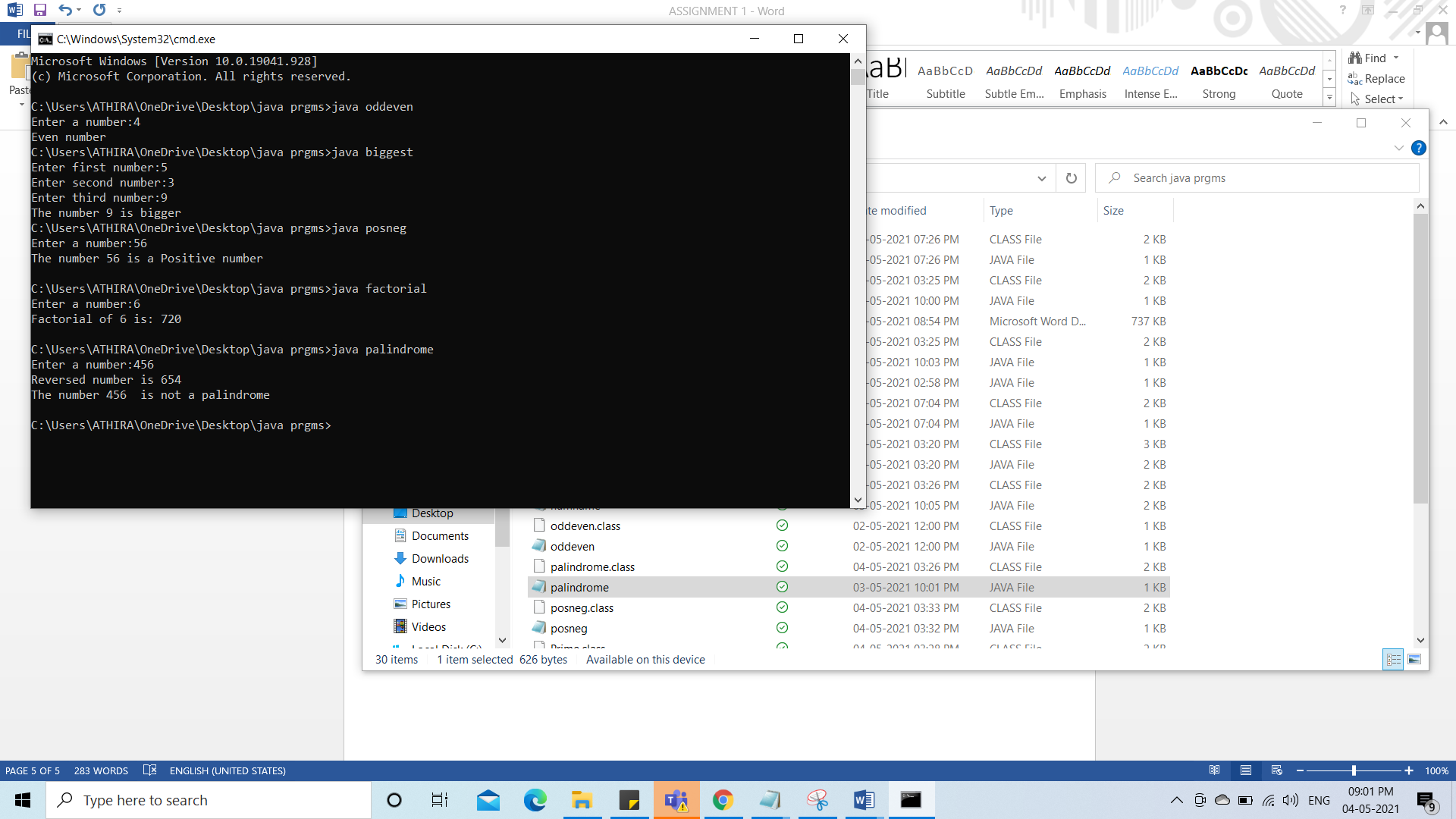
else

System.out.println("The number "+temp+" is not a palindrome");

}

}

Output:



6.NUMBER NAME

Aim: Write a program to display the Number Name of a number between 0 and 9 (use switch statement).

Code:

import java.util.Scanner;

class numname {

public static void main(String[] args) {

Scanner reader=new Scanner(System.in);

System.out.print("Enter a numberbetween 0 and 9:");

int num=reader.nextInt();

switch(num)

{

case 1:

System.out.println("The number name of "+num+" is one");

break;

case 2:

System.out.println("The number name of "+num+" is two");

break;

case 3:

System.out.println("The number name of "+num+" is three");

break;

case 4:

System.out.println("The number name of "+num+" is four");

break;

case 5:

System.out.println("The number name of "+num+" is five");

break;

case 6:

System.out.println("The number name of "+num+" is six");

break;

case 7:

System.out.println("The number name of "+num+" is seven");

break;

case 8:

System.out.println("The number name of "+num+" is eight");

break;

case 9: System.out.println("The number name of "+num+" is nine");

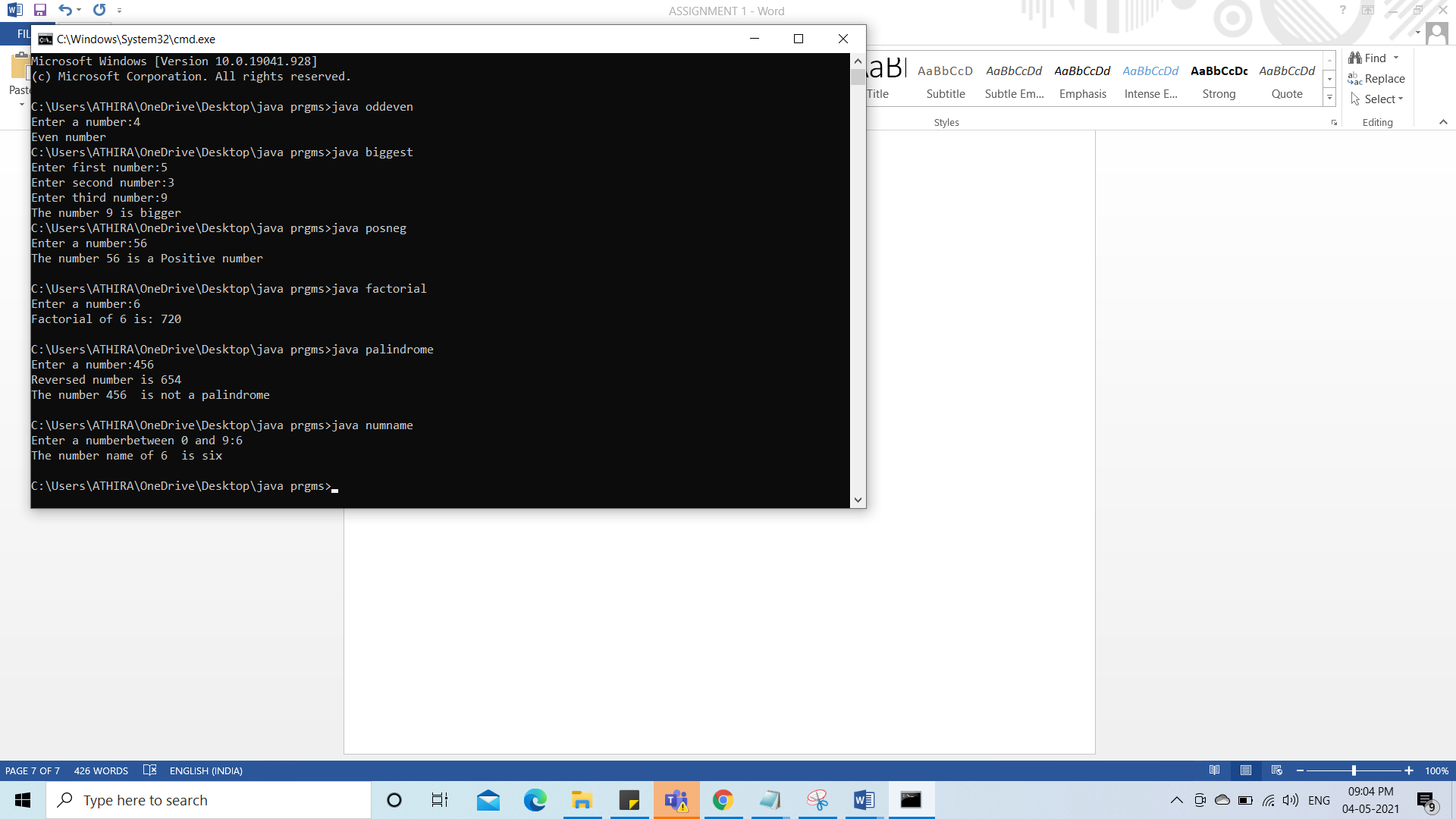
break;

default: System.out.println("Invalid entry");

} }

}

Output:



7.PRIME NUMBER

Aim: Write a program to display prime number upto a limit

Code:

import java.util.Scanner;

class Prime {

public static void main(String[] args) {

Scanner reader=new Scanner(System.in);

System.out.print("Enter a limit:");

int num=reader.nextInt();

System.out.println("Prime numbers are: ");

for(int i=1;i<=num;i++)

{

int p=0;

for(int j=1;j<=i;j++)

{

if(i%j==0)

p++;

}

if(p==2)

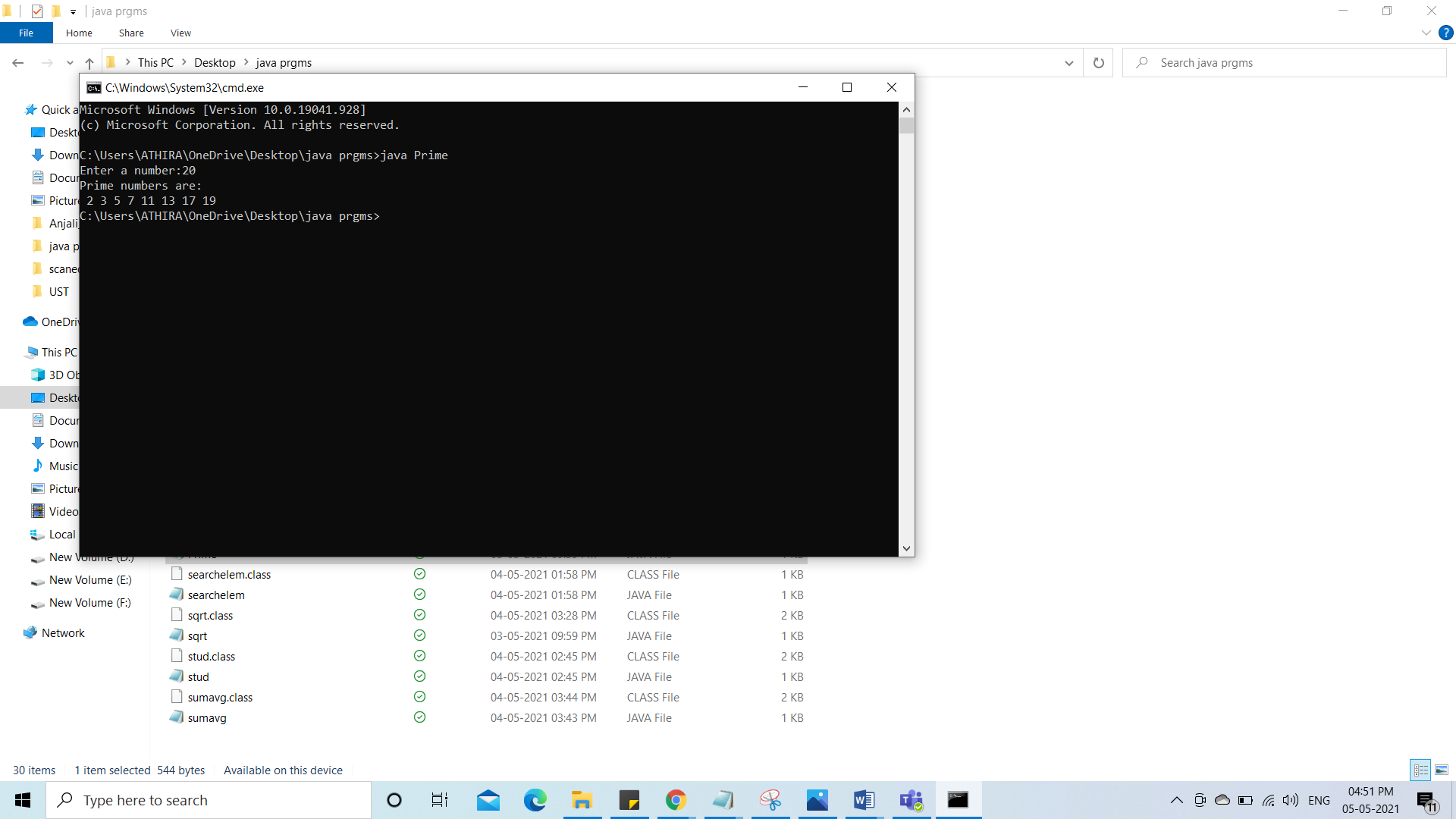
System.out.print(" "+i);

}

}

}

Output:



8.LARGEST IN A LIST

Aim: Write a program to find the largest number in a list

Code:

import java.util.Scanner;

class arraylarge {

public static void main(String[] args) {

int n;

Scanner reader=new Scanner(System.in);

System.out.print("Enter number of elements:");

n=reader.nextInt();

int num[]=new int[n];

System.out.print("Enter array elements:");

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

{

num[i]=reader.nextInt();

}

int max=num[0];

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

{

if(max<num[i])

{

max=num[i];

}

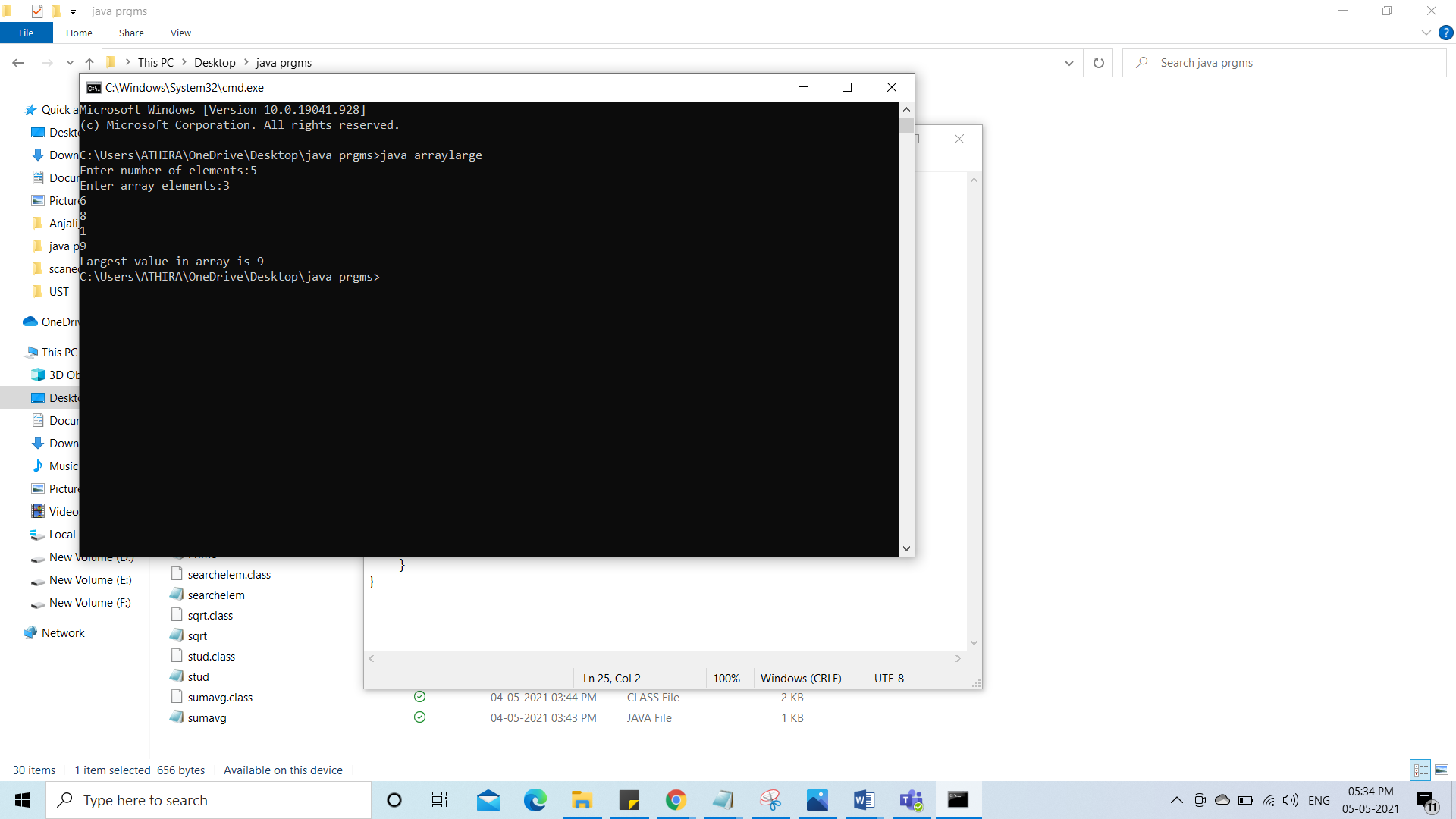
}

System.out.print("Largest value in array is "+max);

}

}

Output:



9.SUM AND AVERAGE OF N NUMBERS

Aim: Write a program to find sum and average on n numbers

Code:

import java.util.Scanner;

class sumavg {

public static void main(String[] args) {

int n,sum=0;

float avg=0;

Scanner reader=new Scanner(System.in);

System.out.print("Enter number of elements:");

n=reader.nextInt();

int num[]=new int[n];

System.out.print("Enter array elements:");

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

{

num[i]=reader.nextInt();

}

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

{

sum=sum+num[i];

}

avg=(float)sum/n;

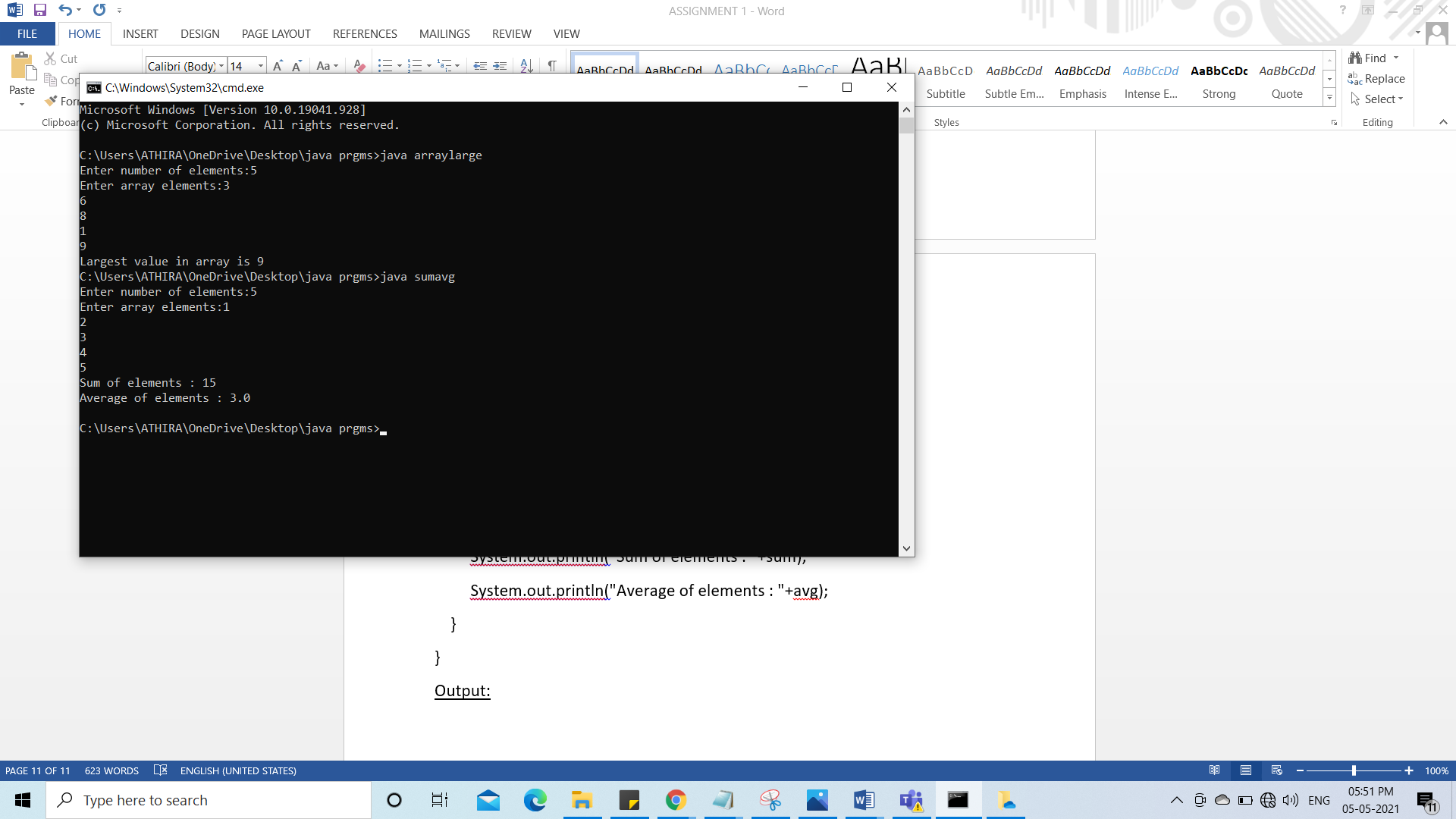
System.out.println("Sum of elements : "+sum);

System.out.println("Average of elements : "+avg);

}

}

Output:



10:NUMBERS IN ASCENDING ORDER

Aim: Write a program to display a list of numbers in ascending order

Code:

import java.util.Scanner;

class arrayasc {

public static void main(String[] args) {

int n, temp;

Scanner reader = new Scanner(System.in);

System.out.print("Enter number of elements:");

n = reader.nextInt();

int a[] = new int[n];

System.out.println("Enter the elements:");

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

{

a[i] = reader.nextInt();

}

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

{

for (int j = i + 1; j < n; j++)

{

if (a[i] > a[j])

{

temp = a[i];

a[i] = a[j];

a[j] = temp;

}

}

}

System.out.print("Ascending Order:");

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

{

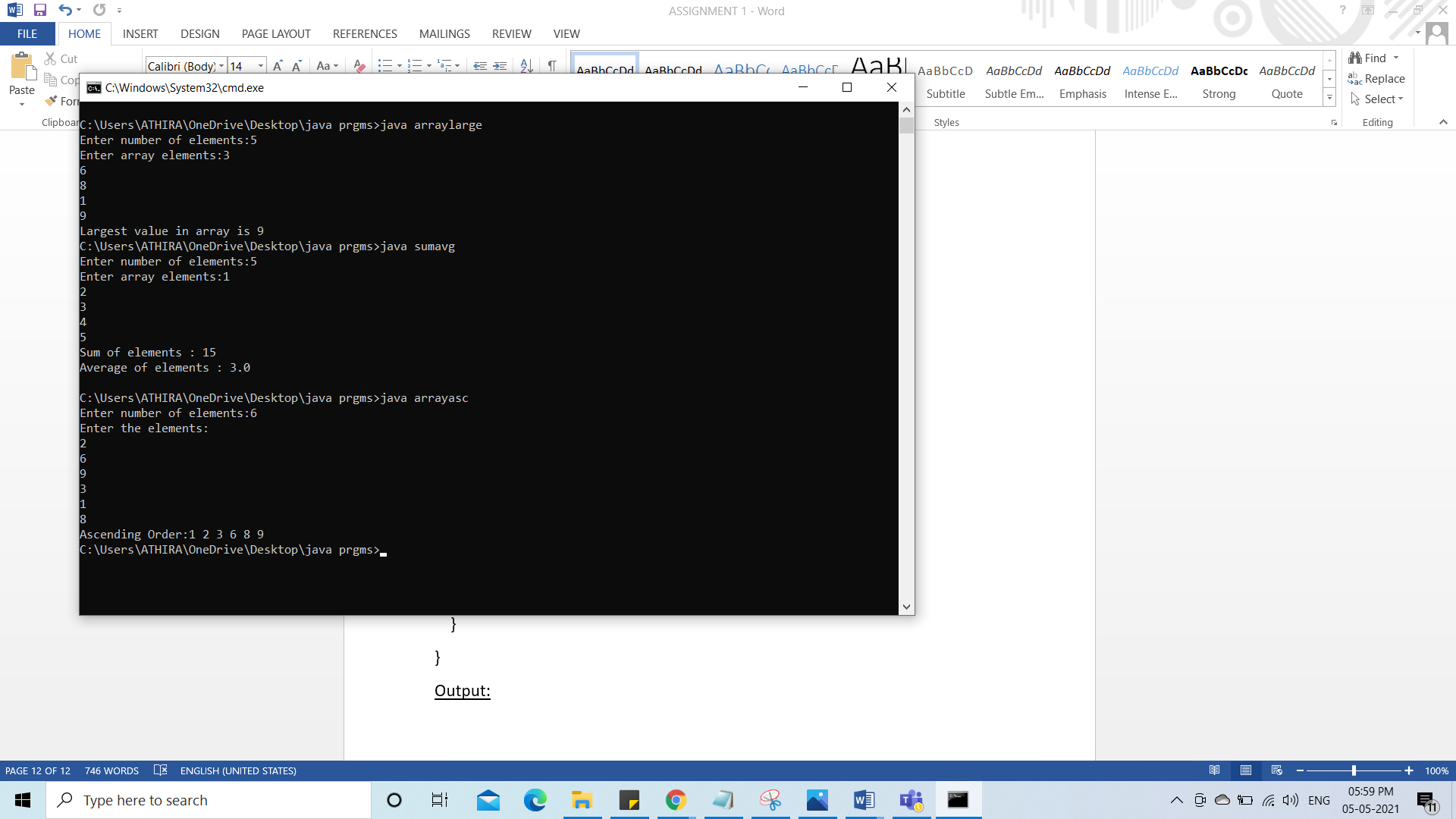
System.out.print(a[i] + " ");

}

}

}

Output:



11. matrix addition.

Aim: Read 2 matrices from the console and perform matrix addition.

Code:

import java.util.Scanner;

class matadd {

public static void main(String[] args) {

int row1,col1,i,j;

Scanner reader = new Scanner(System.in);

System.out.print("Enter number of rows:");

row1 = reader.nextInt();

System.out.print("Enter number of colums:");

col1 = reader.nextInt();

int a[][] = new int[row1][col1];

int b[][]=new int[row1][col1];

int c[][]=new int[row1][col1];

System.out.println("Enter the elements of MAT1:");

for ( i = 0; i < row1; i++) {

for(j=0; j<col1; j++)

{

a[i][j] = reader.nextInt();

}

}

System.out.print("MATRIX 1 :\n");

for ( i = 0; i < row1; i++)

{

for (j =0; j < col1; j++)

{

System.out.print(a[i][j]+" ");

}

System.out.println();

}

System.out.println("Enter the elements of MAT2:");

for ( i = 0; i < row1; i++)

{

for(j=0; j<col1; j++)

{

b[i][j] = reader.nextInt();

}

}

System.out.print("MATRIX 2:\n");

for (i = 0; i < row1; i++)

{

for (j =0; j< col1;j++)

{

System.out.print(b[i][j]+" ");

}

System.out.println();

}

for(i=0;i<row1;i++)

{

for(j=0;j<col1;j++)

{

c[i][j]=a[i][j]+b[i][j];

}

}

System.out.print("\nSUM:\n");

for (i = 0; i <row1; i++)

{

for (j =0;j<col1;j++)

{

System.out.print(c[i][j]+" ");

}

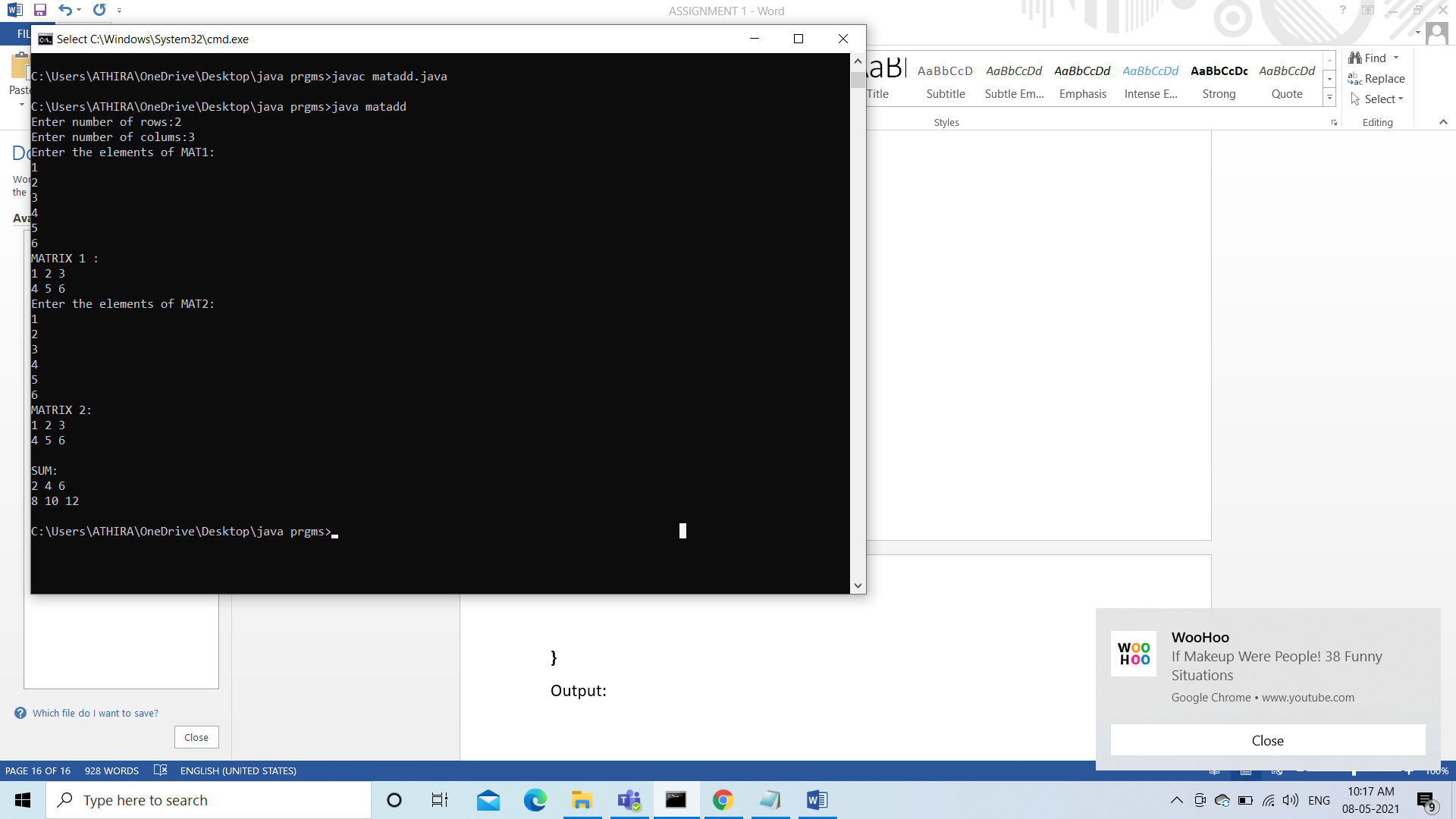
System.out.println();

}

}

}

Output:



12.Matrix Symmetric

Aim: Read a matrix from the console and check whether it is symmetric or not.

Code:

import java.util.Scanner;

class symmat {

public static void main(String[] args) {

int row1,col1,i,j,f=1;

Scanner reader = new Scanner(System.in);

System.out.print("Enter number of rows:");

row1 = reader.nextInt();

System.out.print("Enter number of colums:");

col1 = reader.nextInt();

int a[][] = new int[row1][col1];

int b[][]=new int[row1][col1];

System.out.println("Enter the elements of MAT1:");

for ( i = 0; i < row1; i++)

{

for(j=0; j<col1; j++)

{

a[i][j] = reader.nextInt();

}

}

System.out.print("MATRIX 1 :\n");

for ( i = 0; i < row1; i++)

{

for (j =0; j < col1; j++)

{

System.out.print(a[i][j]+" ");

}

System.out.println();

}

if(row1==col1)

{

for (i = 0; i < row1; i++)

{

for (j =0; j< col1;j++)

{

b[i][j]=a[j][i];

}

}

System.out.print("TRANSPOSE:\n");

for ( i = 0; i < row1; i++)

{

for (j =0; j < col1; j++)

{

System.out.print(b[i][j]+" ");

}

System.out.println();

}

}

for(i=0;i<row1;i++)

{

for(j=0;j<col1;j++)

{

if(a[i][j]!=b[i][j])

{

f=0;

break;

}

}

}

if(f==0)

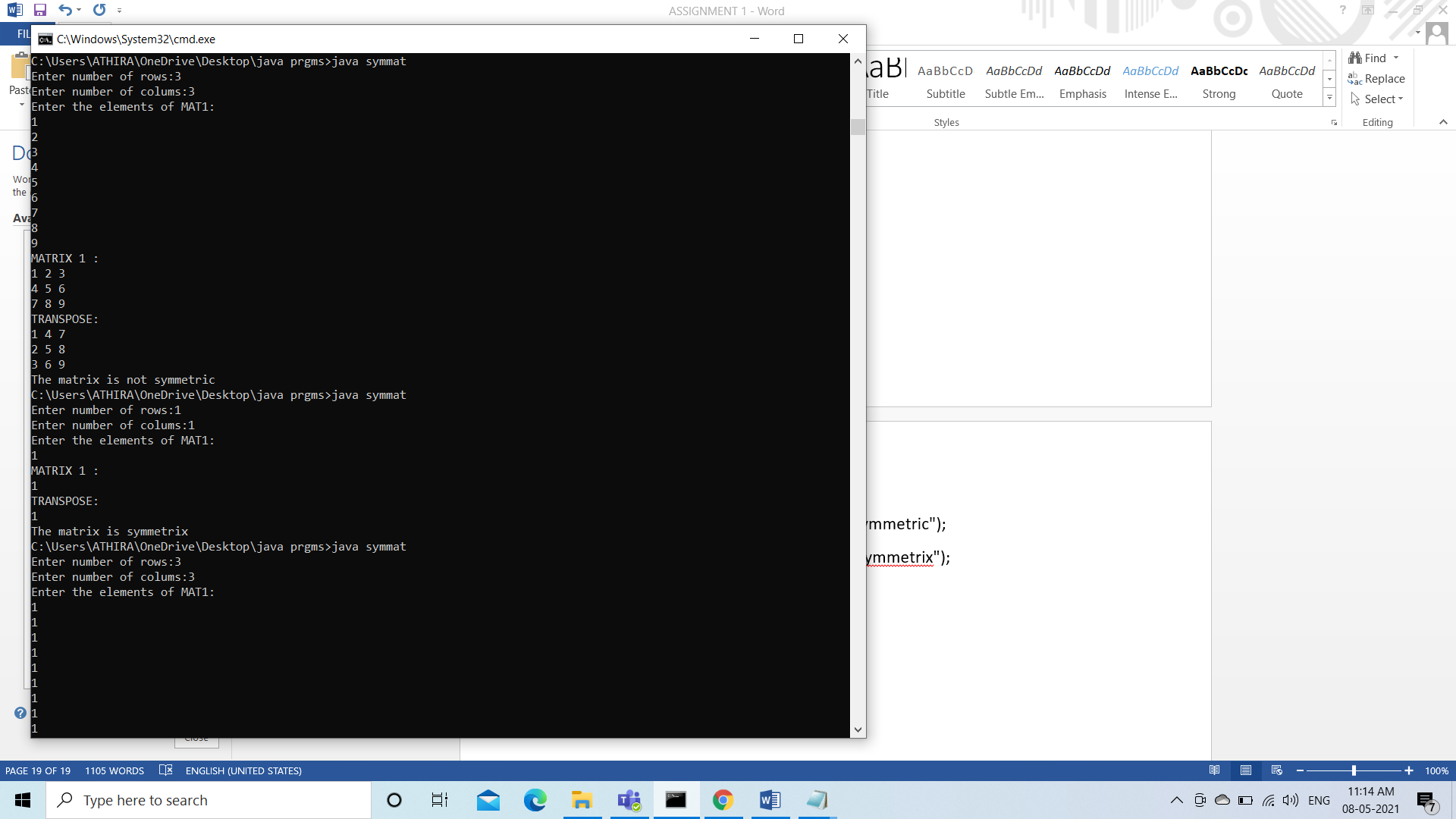
System.out.print("The matrix is not symmetric");

else System.out.print("The matrix is symmetrix");

}

}

Output:



13.Search an element

Aim: Search an element in an array.

Code:

import java.util.Scanner;

class searchelem {

public static void main(String[] args) {

int n,c=0;

Scanner reader=new Scanner(System.in);

System.out.print("Enter number of elements:");

n=reader.nextInt();

int num[]=new int[n];

System.out.println("Enter array elements:");

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

{

num[i]=reader.nextInt();

}

System.out.print("Enter the element to be searched: ");

int elem=reader.nextInt();

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

{

if(elem==num[i])

{

c++;

}

}

if(c==0)

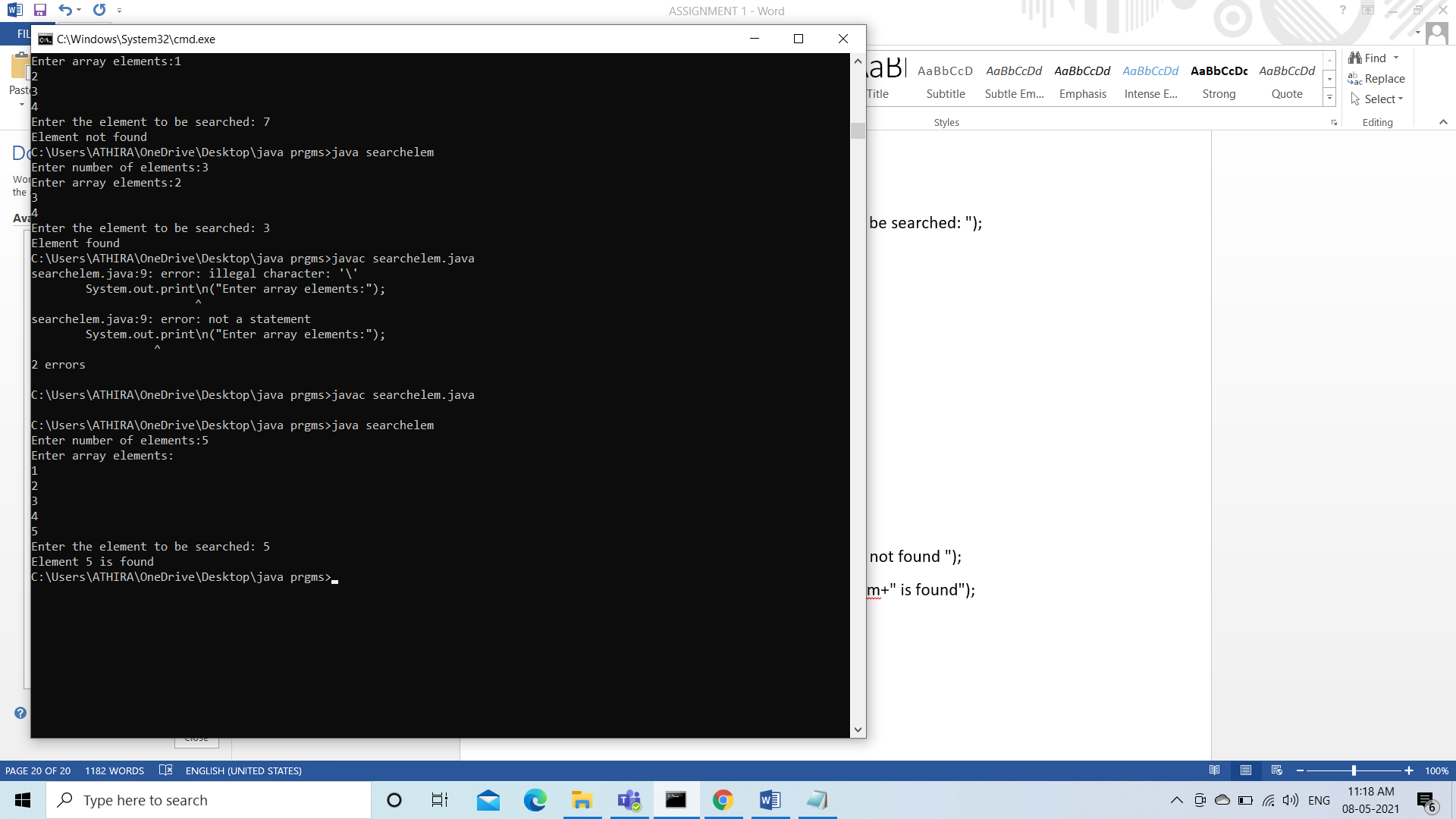
System.out.print("Element "+elem+" is not found ");

else System.out.print("Element " +elem+" is found "+c+” times” );

}

}

Output:



14.Student details

Aim: Program to create a class Student having attributes name, roll, branch, mark, grade of datatype String, integer, array of character, double/float, character respectively. Read the student information and display the same.

Code:

import java.util.Scanner;

class stud {

public static void main(String[] args) {

String name;

int roll,i;

char[] branch;

double mark;

char grade;

Scanner reader=new Scanner(System.in);

System.out.print("Enter student name:");

name= reader.nextLine();

System.out.print("Enter roll number:");

roll=reader.nextInt();

System.out.print("Enter branch:");

branch=reader.next().toCharArray();

System.out.print("Enter student mark:");

mark=reader.nextDouble();

System.out.print("Enter student grade:");

grade=reader.next().charAt(0);

System.out.println("\n\nStudent details:\nName: "+name+"\nRoll number: "+roll+

"\nBranch name: ");

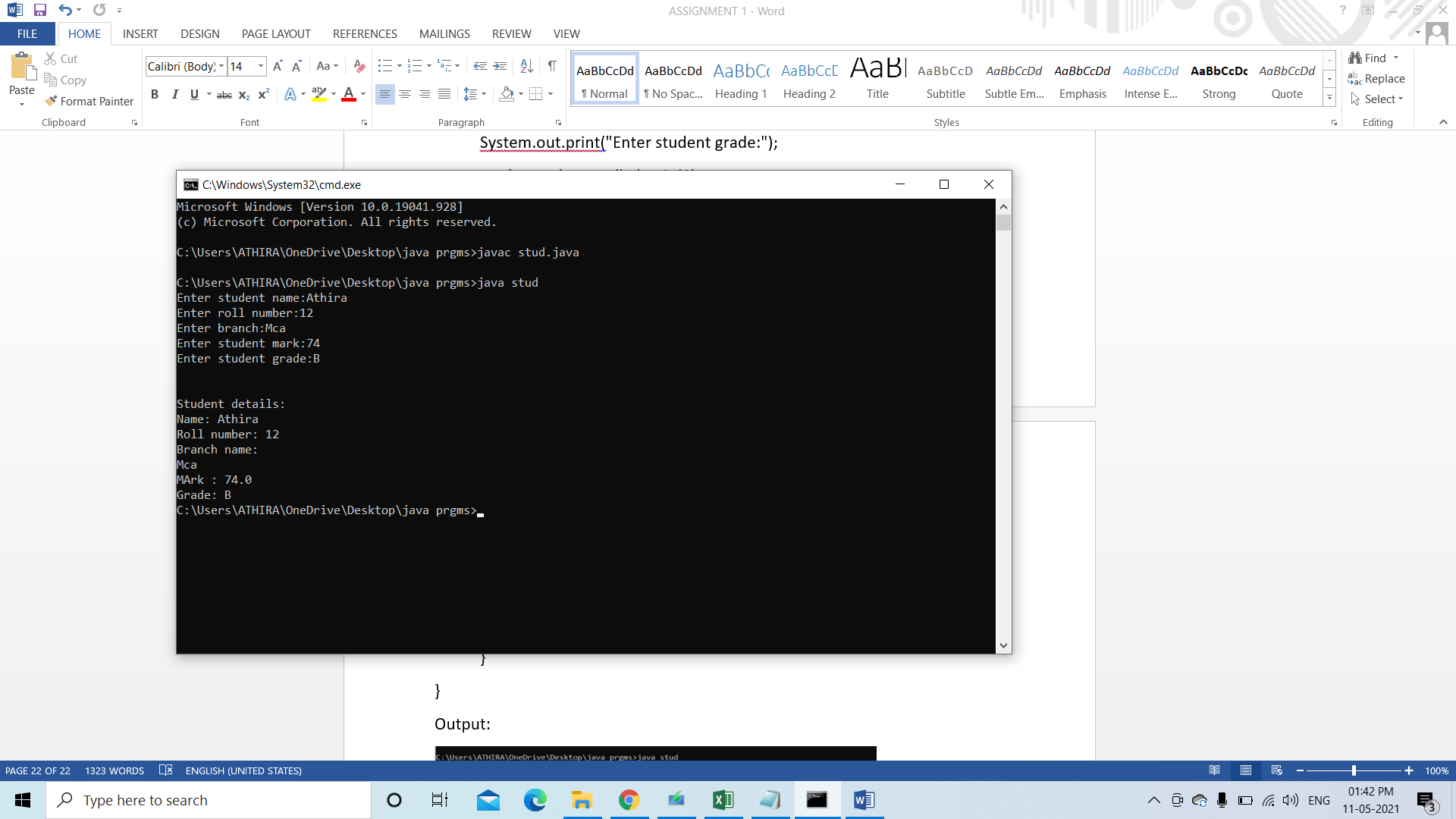
for(i=0;i<branch.length;i++)

System.out.print(branch[i]);

System.out.print("\nMArk : "+mark+"\nGrade: "+grade);

} }

Output:



15.Circle

Aim: Create a class Circle with fields x co-ordinate, y co-ordinate and radius , Also having member functions area() and circumference(). Write a program to compute the circumference and area of the circle when user enters the centre and radius of the circle.

Code:

import java.util.Scanner;  
class circle{  
 double c,a;  
 public void area(int z)  
 {  
 a=3.14\*z\*z;  
 System.out.print("Area: "+a);  
 }  
 public void circumference(int z)  
 {  
 c= 2\*3.14\*z;  
 System.out.print("Circumference: "+c);  
 }  
  
 public static void main(String args[])  
 {  
 int cen,r;  
 Scanner reader = new Scanner(System.in);  
 System.out.print("Enter radius: ");  
 r=reader.nextInt();  
 System.out.print("Enter center: ");  
 cen=reader.nextInt();  
 circle acircle=new circle();  
 acircle.area(r);   
 acircle.circumference(r);  
 }  
}

16:Product Lowest

Aim: Define a class ‘product’ with data members pcode, pname and price. Create 3 objects of the class and find the product having the lowest price.

Code:

import java.util.Scanner;  
 class product{  
 String pcode,pname;  
 float price;  
 product(){  
 pcode=" ";  
 pname=" ";  
 price=0;  
 }  
 public void input(){  
 Scanner reader=new Scanner(System.*in*);  
 System.*out*.print("Enter product code :");  
 pcode=reader.nextLine();  
 System.*out*.print("Enter product name :");  
 pname=reader.nextLine();  
 System.*out*.print("Enter product price :");  
 price=reader.nextFloat();  
 }  
 public product lowest(product p1,product p2){  
 product temp=new product();  
 if(this.price<p1.price && this.price<p2.price)  
 temp=this;  
 else if(p1.price<this.price && p1.price<p2.price)  
 temp=p1;  
 else if(p2.price<this.price && p2.price<p1.price)  
 temp=p2;  
 return temp;  
 }  
 public void output() {  
 System.*out*.println("PRODUCT CODE:"+pcode);  
 System.*out*.println("PRODUCT NAME:"+pname);  
 System.*out*.println("PRODUCT PRICE:"+price+"\n");  
 }  
 };  
 public class productlowest {  
 public static void main(String[] args) {  
 product p1 = new product();  
 product p2 = new product();  
 product p3 = new product();  
 product result = new product();  
 System.*out*.print("Enter details of product1\n ");  
 p1.input();  
 System.*out*.print("\nEnter details of product2\n ");  
 p2.input();  
 System.*out*.print("\nEnter details of product3\n ");  
 p3.input();  
 result = p3.lowest(p1, p2);  
 System.*out*.println("\nDetails of Product1 ");  
 p1.output();  
 System.*out*.println("Details of Product2");  
 p2.output();  
 System.*out*.println("Details of Product3");  
 p3.output();  
 System.*out*.println("Product with Lowest Price");  
 result.output();  
  
 }  
 }

17.Complex Number

Aim: Add complex numbers

Code:

import java.util.Scanner;  
  
 class complex{  
 float real;  
 float imag;  
 complex(){  
 real=0;  
 imag=0;  
 }  
 public void input(){  
 Scanner reader=new Scanner(System.*in*);  
 System.*out*.print("Enter Real part :");  
 real=reader.nextFloat();  
 System.*out*.print("Enter imaginary part :");  
 imag=reader.nextFloat();  
 }  
 public complex add(complex obj){  
 complex c=new complex();  
 c.real=real+obj.real;  
 c.imag=imag+obj.imag;  
 return c;  
 }  
 public void output() {  
 if (imag < 0)  
 System.*out*.print("Complex number1: " + real + " " + imag + "i");  
 else  
 System.*out*.print("Complex number2: " + real + " " + imag + "i");  
 }  
 };  
 public class complexadd {  
 public static void main(String[] args){  
  
 complex c1=new complex();  
 complex c2=new complex();  
 complex result=new complex();  
 System.*out*.print("Enter first complex number\n ");  
 c1.input();  
 System.*out*.print("Enter second complex number\n ");  
 c2.input();  
 result = c1.add(c2);  
 result.output();  
 } }

18.Cpu

Aim: Create CPU with attribute price. Create inner class Processor (no. of cores, manufacturer) and static nested class RAM (memory, manufacturer). Create an object of CPU and print information of Processor and RAM.

Code:

import java.util.Scanner;  
class cpuinfo{  
 int price;  
 class Processor{  
 String Manufacturer, cores;  
 }  
 static class Ram{  
 String Manufacturer;  
 int mem;  
 }  
}  
class Computer{  
 public static void main(String args[]){  
 cpuinfo cpu = new cpuinfo();  
 cpuinfo.Processor p=cpu.new Processor();  
 cpuinfo.Ram r=new cpuinfo.Ram();  
 Scanner in = new Scanner(System.*in*);  
 System.*out*.println("Enter The CPU Details\n");  
 System.*out*.print("Enter RAM Manufacture Name : ");  
 r.Manufacturer=in.next();  
 System.*out*.print("Enter RAM Memory : ");  
 r.mem=in.nextInt();  
 System.*out*.print("Enter Processor Manufacture Name : ");  
 p.Manufacturer=in.next();  
 System.*out*.print("Enter No.Of.Cores : ");  
 p.cores=in.next();  
 System.*out*.print("Enter CPU Price : ");  
 cpu.price=in.nextInt();  
 System.*out*.println("\nCPU Details");  
 System.*out*.println("\nRAM Manufacture Name : "+ r.Manufacturer);  
 System.*out*.println("RAM Memory : "+ r.mem);  
 System.*out*.println("Processor Manufacture Name : "+ p.Manufacturer);  
 System.*out*.println("No.Of.Cores : "+ p.cores);  
 System.*out*.println("CPU Price : "+ cpu.price);

} }

19.Class Employee

Aim: Program to create a class for Employee having attributes eNo, eName eSalary. Read n employ information and Search for an employee given eNo, using the concept of Array of Objects.

Code:

import java.util.Scanner;  
public class Employeeinfo {  
 int eno;  
 String ename;  
 float esalary;  
 Employeeinfo(){  
 eno=0;  
 ename=" ";  
 esalary=0;  
 }  
 public void input() {  
 Scanner reader = new Scanner(System.*in*);  
 System.*out*.print("Enter the employee number : ");  
 eno = reader.nextInt();  
 System.*out*.print("Enter the employee name : ");  
 ename = reader.next();  
 System.*out*.print("Enter the employee salary : ");  
 esalary = reader.nextFloat();  
 }  
 public void output() {  
 System.*out*.println("Employee id = " + eno);  
 System.*out*.println("Employee name = " + ename);  
 System.*out*.println("Employee salary = " + esalary);  
 }  
public static void main(String[] args) {  
 int n;  
 Scanner reader = new Scanner(System.*in*);  
 System.*out*.print("Enter the number of Employees: ");  
 n = reader.nextInt();  
 Employeeinfo e[] = new Employeeinfo[n];  
  
 for (int i = 0; i < n; i++) {  
 e[i] = new Employeeinfo();  
 e[i].input();  
 System.*out*.print("\n");  
 }  
 System.*out*.print("Enter employee number to be searched :");  
 int elem = reader.nextInt();  
 int f=0;  
 System.*out*.println("Employee Details");  
 for (int i = 0; i < n; i++) {  
 if (e[i].eno==elem) {  
 f=1;  
 break;  
 }  
 }  
 if(f==0)  
 System.*out*.print("Employee not present");  
 else  
 {  
 for (int i = 0; i < n; i++){  
 if (e[i].eno==elem)  
 e[i].output();  
 }  
 }  
 }  
}

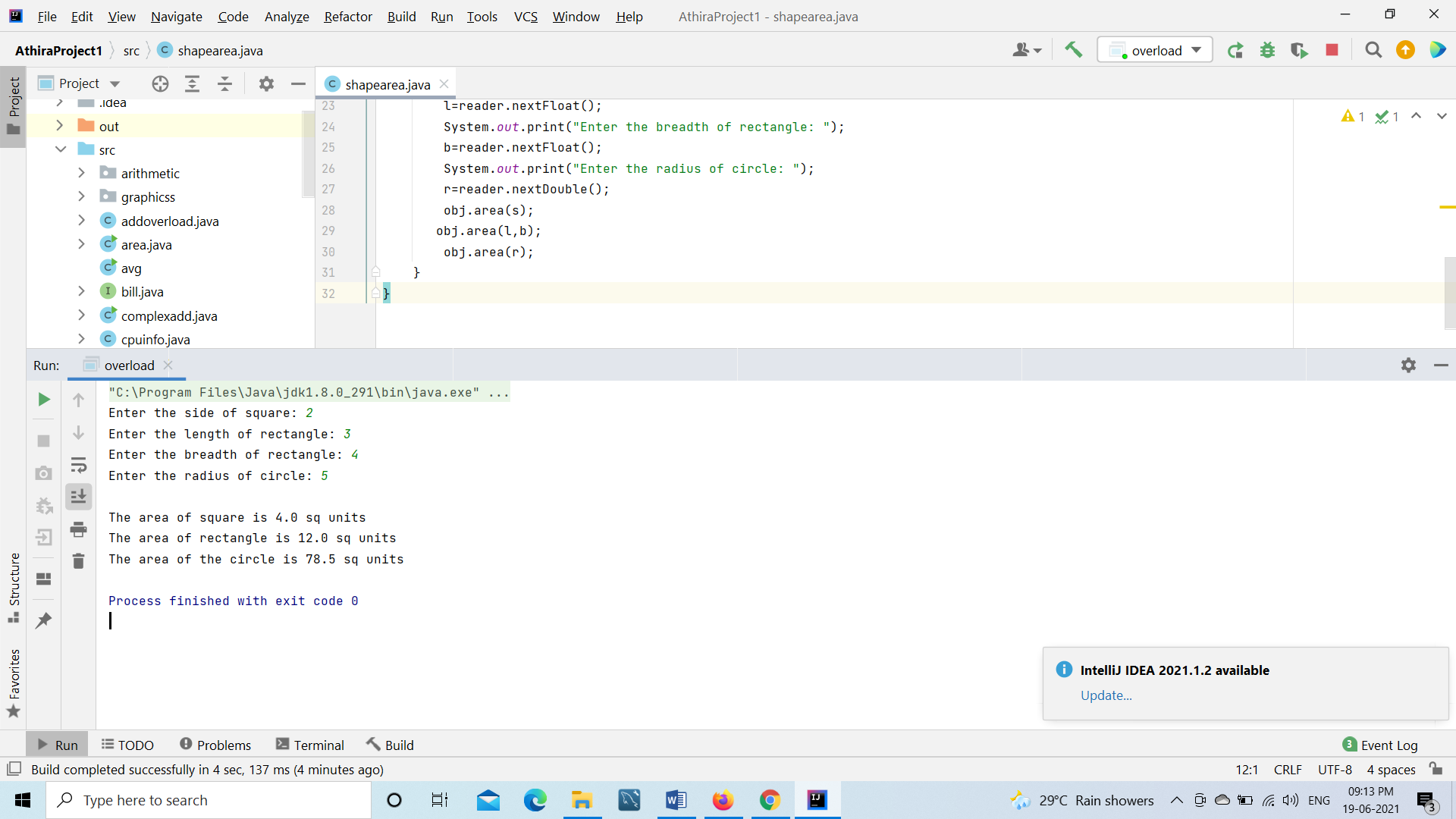
20.Area overload

Aim: Area of different shapes using overloaded functions

Code:

import java.util.Scanner;  
public class shapearea {  
 void area(float x){  
 System.*out*.print("\nThe area of square is "+Math.*pow*(x,2)+" sq units\n");  
 }  
 void area(float x,float y){  
 System.*out*.print("The area of rectangle is "+x\*y+" sq units\n");  
 }  
 void area(double x){  
 double z=3.14\*x\*x;  
 System.*out*.print("The area of the circle is "+z+" sq units\n" );  
 }  
}  
class overload{  
 public static void main(String args[]){  
 float s,l,b;  
 double r;  
 Scanner reader=new Scanner(System.*in*);  
 shapearea obj=new shapearea();  
 System.*out*.print("Enter the side of square: ");  
 s=reader.nextFloat();  
 System.*out*.print("Enter the length of rectangle: ");  
 l=reader.nextFloat();  
 System.*out*.print("Enter the breadth of rectangle: ");  
 b=reader.nextFloat();  
 System.*out*.print("Enter the radius of circle: ");  
 r=reader.nextDouble();  
 obj.area(s);  
 obj.area(l,b);  
 obj.area(r);  
 }  
}

OUTPUT:



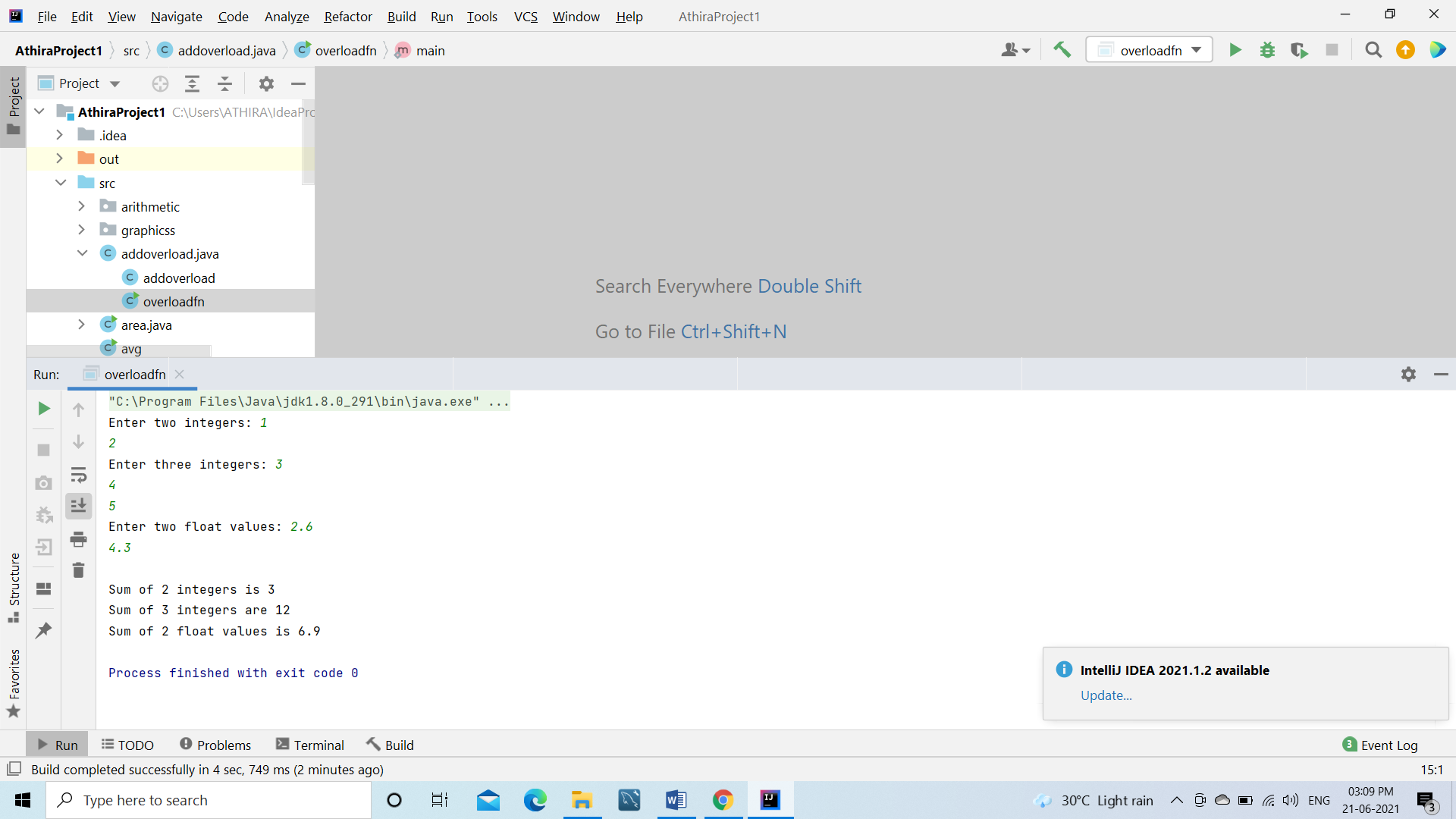
21.ADD OVERLOAD

Aim:Write a Java program to perform addition on 2 integers, 3 integers and 2 float values using overloaded functions.

Code:

import java.util.Scanner;  
public class addoverload {  
 void add(int x,int y){  
 System.*out*.print("\n" +  
 "Sum of 2 integers is "+(x+y)+" \n");  
 }  
 void add(int x,int y, int z){  
 System.*out*.print("Sum of 3 integers are "+(x+y+z)+" \n");  
 }  
 void add(float x,float y){  
 System.*out*.print("Sum of 2 float values is "+(x+y)+" \n" );  
 }  
 }  
 class overloadfn{  
 public static void main(String args[]){  
 int a,b,c,d,e;  
 float m,n;  
 Scanner reader=new Scanner(System.*in*);  
 addoverload obj=new addoverload();  
 System.*out*.print("Enter two integers: ");  
 a=reader.nextInt();  
 b=reader.nextInt();  
 System.*out*.print("Enter three integers: ");  
 c=reader.nextInt();  
 d=reader.nextInt();  
 e=reader.nextInt();  
 System.*out*.print("Enter two float values: ");  
 m=reader.nextFloat();  
 n=reader.nextFloat();  
 obj.add(a,b);  
 obj.add(c,d,e);  
 obj.add(m,n);  
 }  
 }

Output:



22. Class employee

Aim: Create a class ‘Employee’ with data members Empid, Name, Salary, Address and constructors to initialize the data members. Create another class ‘Teacher’ that inherits the properties of class Employee and contains its own data members department, Subjects taught and constructors to initialize these data members and also include a display function to display all the data members. Use an array of objects to display details of N teachers.

Code:

import java.util.Scanner;  
public class employee {  
 int empid;  
 String name;  
 float salary;  
 String address;  
 public employee() {  
 empid = 0;  
 name = " ";  
 salary = 0;  
 address = " ";  
 }  
 public void input() {  
 Scanner reader = new Scanner(System.*in*);  
 System.*out*.print("Enter the employee number : ");  
 empid = reader.nextInt();  
 System.*out*.print("Enter the employee name : ");  
 name = reader.next();  
 System.*out*.print("Enter the employee salary : ");  
 salary = reader.nextFloat();  
 System.*out*.print("Enter the employee address : ");  
 address = reader.next();  
 }  
}  
class teacher extends employee {  
 String dept,sub;  
 *//String sub[];* public teacher(){  
 dept=" ";  
 sub=" ";  
 }  
 public void data() {  
 *//int m;* Scanner reader = new Scanner(System.*in*);  
 System.*out*.print("Enter the department : ");  
 dept = reader.next();  
 *//System.out.print("Enter the number of subjects: ");  
 //m=reader.nextInt();  
 //String[] sub=new String[50];* System.*out*.print("Enter the subject taught : ");  
 sub=reader.next();  
 *//for (int i = 0; i<m; i++){  
 // sub[i] = reader.nextLine();  
 // }* }  
 public void output() {  
 System.*out*.println("Employee id = " + empid);  
 System.*out*.println("Employee name = " + name);  
 System.*out*.println("Employee salary = " + salary);  
 System.*out*.println("Employee address = " + address);  
 System.*out*.println("Employee department = " + dept);  
 System.*out*.println("Subject taught = "+sub);  
 *//for (int i=0; i<sub.length; i++) {  
 // System.out.println(sub[i]+" ");  
 //}* }  
 public static void main(String[] args) {  
 int n;  
 Scanner reader = new Scanner(System.*in*);  
 System.*out*.print("Enter the number of Employees: ");  
 n = reader.nextInt();  
 teacher t[] = new teacher[n];  
 for (int i = 0; i < n; i++) {  
 t[i] = new teacher();  
 t[i].input();  
 t[i].data();  
 System.*out*.print("\n");  
 }  
 System.*out*.println("EMPLOYEE DETAILS: ");  
 for (int i = 0; i < n; i++) {  
 t[i].output();  
 System.*out*.print("\n");  
 }  
 }  
}

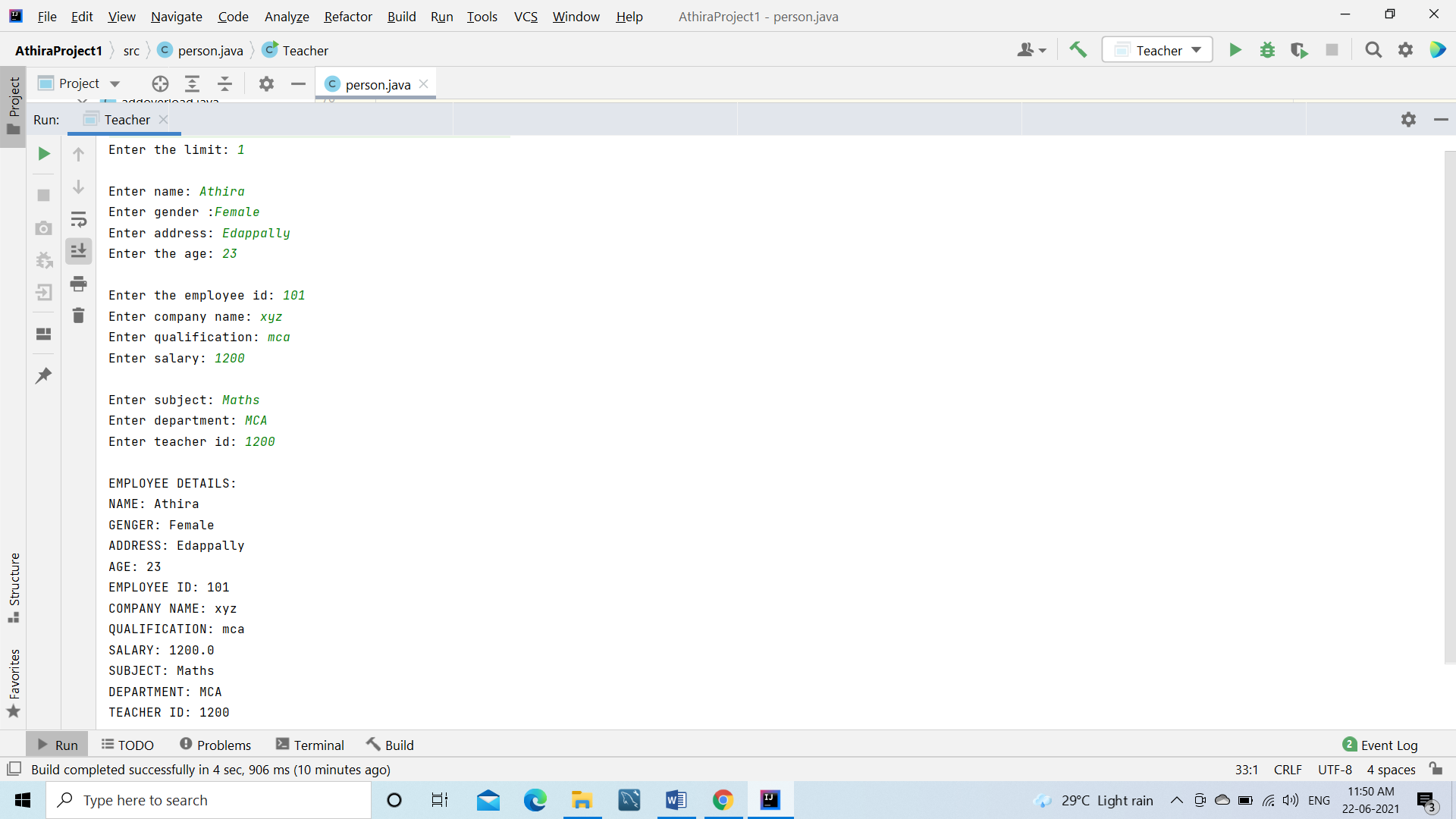
23.Class person

Aim: Create a class ‘Person’ with data members Name, Gender, Address, Age and a constructor to initialize the data members and another class ‘Employee’ that inherits the properties of class Person and also contains its own data members like Empid, Company\_name, Qualification, Salary and its own constructor. Create another class ‘Teacher’ that inherits the properties of class Employee and contains its own data members like Subject, Department, Teacher id and also contains constructors and methods to display the data members. Use an array of objects to display details of N teachers.

Code:

import java.util.Scanner;  
public class person {  
 String name,gender,address;  
 int age;  
 public person(){  
 name=" ";  
 gender=" ";  
 address=" ";  
 age=0;  
 }  
 public void pinput(){  
 Scanner reader=new Scanner(System.*in*);  
 System.*out*.print("\nEnter name: ");  
 name=reader.next();  
 System.*out*.print("Enter gender :");  
 gender=reader.next();  
 System.*out*.print("Enter address: ");  
 address=reader.next();  
 System.*out*.print("Enter the age: ");  
 age=reader.nextInt();  
 }  
}  
 class Employee extends person{  
 int empid;  
 String comp\_name,qualification;  
 float sal;  
 public Employee(){  
 empid=0;  
 comp\_name=" ";  
 qualification=" ";  
 sal=0;  
 }  
 public void einput(){  
 Scanner reader=new Scanner(System.*in*);  
 System.*out*.print("Enter the employee id: ");  
 empid=reader.nextInt();  
 System.*out*.print("Enter company name: ");  
 comp\_name=reader.next();  
 System.*out*.print("Enter qualification: ");  
 qualification=reader.next();  
 System.*out*.print("Enter salary: ");  
 sal=reader.nextFloat();  
 }  
}  
class Teacher extends Employee{  
 String subject,dept;  
 int tid;  
 public Teacher(){  
 subject=" ";  
 dept=" ";  
 tid=0;  
 }  
 public void tinput(){  
 Scanner reader=new Scanner(System.*in*);  
 System.*out*.print("Enter subject: ");  
 subject=reader.next();  
 System.*out*.print("Enter department: ");  
 dept=reader.next();  
 System.*out*.print("Enter teacher id: ");  
 tid=reader.nextInt();  
 }  
 public void output(){  
 System.*out*.println("\nNAME: "+name);  
 System.*out*.println("GENGER: "+gender);  
 System.*out*.println("ADDRESS: "+address);  
 System.*out*.println("AGE: "+age);  
 System.*out*.println("EMPLOYEE ID: "+empid);  
 System.*out*.println("COMPANY NAME: "+comp\_name);  
 System.*out*.println("QUALIFICATION: "+qualification);  
 System.*out*.println("SALARY: "+sal);  
 System.*out*.println("SUBJECT: "+subject);  
 System.*out*.println("DEPARTMENT: "+dept);  
 System.*out*.println("TEACHER ID: "+tid+"\n");  
 }  
 public static void main(String[] args) {  
 int n;  
 Scanner reader = new Scanner(System.*in*);  
 System.*out*.print("Enter the limit: ");  
 n = reader.nextInt();  
 Teacher t[] = new Teacher[n];  
 for (int i = 0; i < n; i++) {  
 t[i] = new Teacher();  
 t[i].pinput();  
 System.*out*.print("\n");  
 t[i].einput();  
 System.*out*.print("\n");  
 t[i].tinput();  
 }  
 for (int i=0;i<n;i++){  
 System.*out*.print("\nEMPLOYEE DETAILS: ");  
 t[i].output();  
 }  
 }  
}

Output:



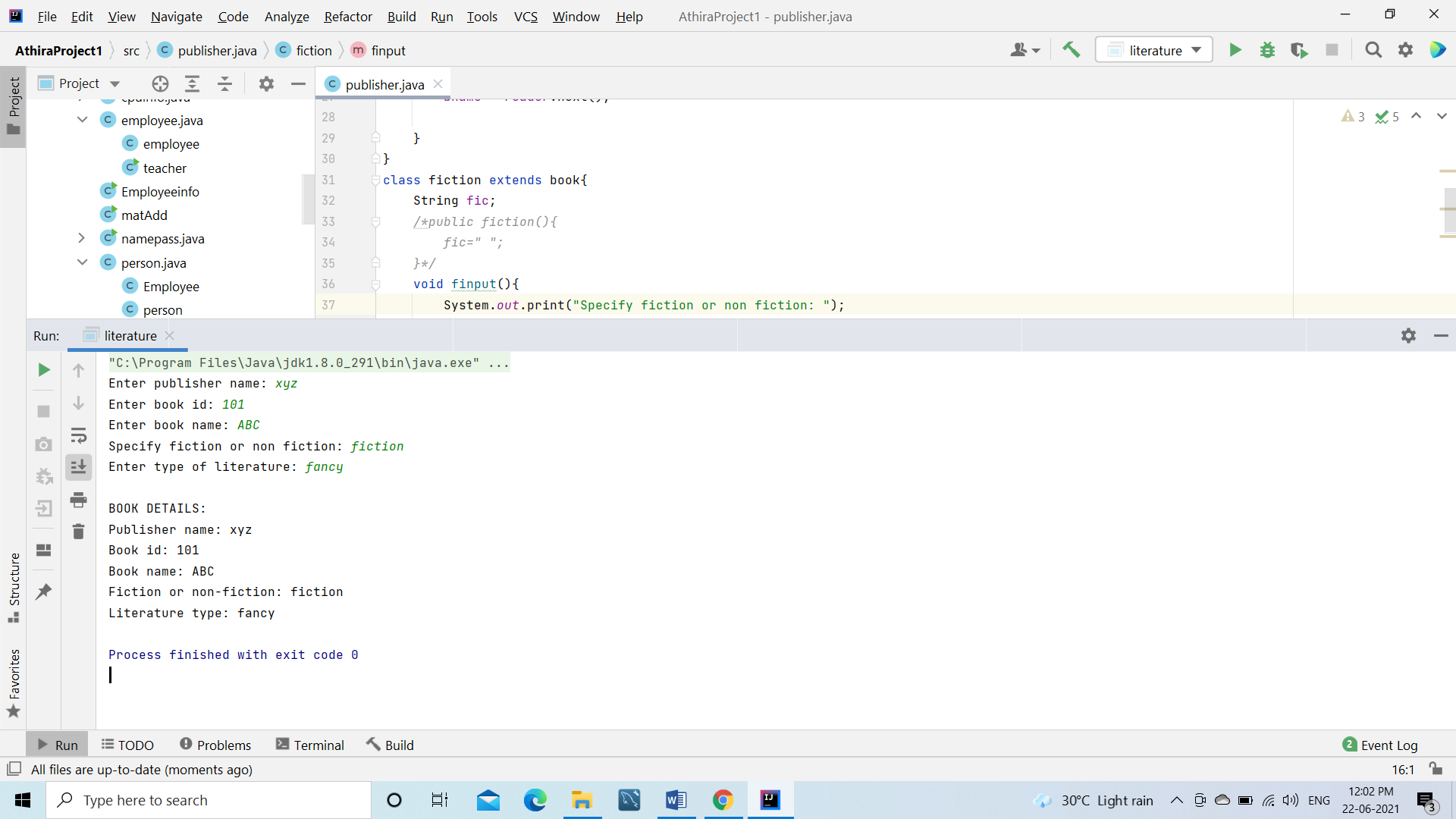
24.Class publisher,book,fiction,literature

Aim: Write a program that has class Publisher, Book, Literature and Fiction. Read the information and print the details of books from either the category, using inheritance.

Code:

import java.util.Scanner;  
public class publisher {  
 Scanner reader=new Scanner(System.*in*);  
 String name;  
 public publisher(){  
 name=" ";  
 }  
 public void input(){  
 System.*out*.print("Enter publisher name: ");  
 name=reader.next();  
 }  
}  
class book extends publisher {  
 String bname;  
 int bid;  
 public book() {  
 bname = " ";  
 bid = 0;  
 }  
 public void binput() {  
 System.*out*.print("Enter book id: ");  
 bid = reader.nextInt();  
 System.*out*.print("Enter book name: ");  
 bname = reader.next();  
 }  
}  
class fiction extends book{  
 String fic;  
 public fiction(){  
 fic=" ";  
 }  
 void finput(){  
 System.*out*.print("Specify fiction or non fiction: ");  
 fic=reader.next();  
 }  
}  
class literature extends book {  
 String lit;  
  
 public literature() {  
 lit = " ";  
 }  
 void linput() {  
 System.*out*.print("Enter type of literature: ");  
 lit = reader.next();  
 }  
 public static void main(String[] args) {  
 literature i = new literature();  
 fiction f=new fiction();  
 i.input();  
 i.binput();  
 f.finput();  
 i.linput();  
 System.*out*.println("\nBOOK DETAILS: ");  
 System.*out*.println("Publisher name: " + i.name);  
 System.*out*.println("Book id: " + i.bid);  
 System.*out*.println("Book name: " + i.bname);  
 System.*out*.println("Fiction or non-fiction: " + f.fic);  
 System.*out*.println("Literature type: " + i.lit);  
 }  
}

Output:



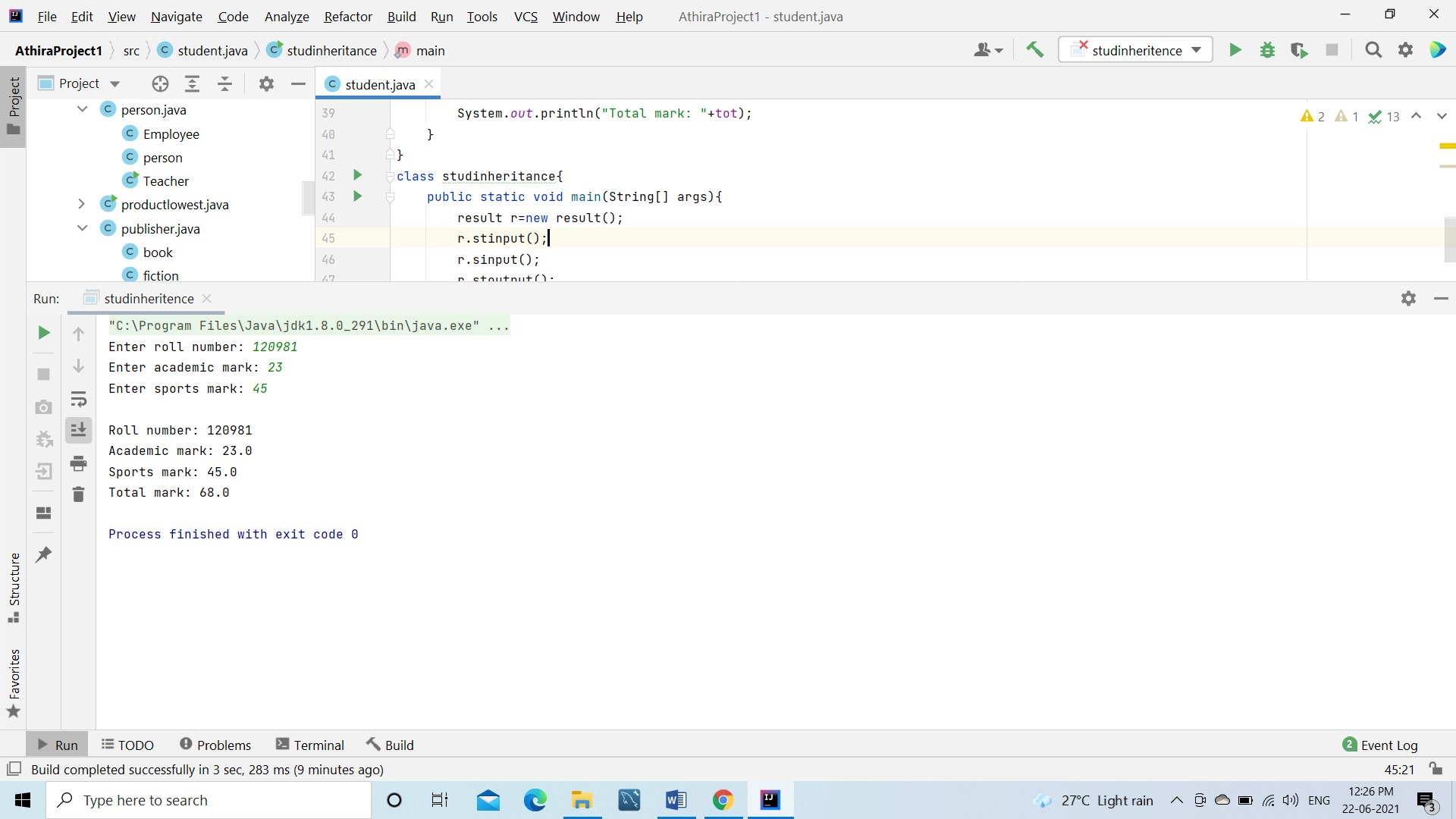
25.Classes student,sports and result

Aim:Create classes Student and Sports. Create another class Result inherited from Student and Sports. Display the academic and sports score of a student.

Code:

import java.util.Scanner;  
interface sports {  
 *// float smark=0;* void sinput();  
 void soutput();  
}  
class student {  
 Scanner reader=new Scanner(System.*in*);  
 int rolln;  
 float amark;  
 student(){  
 rolln=0;  
 amark=0;  
 }  
 void stinput(){  
 System.*out*.print("Enter roll number: ");  
 rolln=reader.nextInt();  
 System.*out*.print("Enter academic mark: ");  
 amark=reader.nextFloat();  
 }  
 void stoutput(){  
 System.*out*.println("\nRoll number: "+rolln);  
 System.*out*.println("Academic mark: "+amark);  
 }  
}  
class result extends student implements sports{  
 Scanner reader=new Scanner(System.*in*);  
 float tot,smark;  
 public void sinput(){  
 System.*out*.print("Enter sports mark: ");  
 smark = reader.nextFloat();  
 }  
 public void soutput() {  
 System.*out*.println("Sports mark: "+smark);  
 tot=amark+smark;  
 System.*out*.println("Total mark: "+tot);  
 }  
}  
class studinheritence{  
 public static void main(String[] args){  
 result r=new result();  
 r.stinput();  
 r.sinput();  
 r.stoutput();  
 r.soutput();  
 }  
}

Output:



26.Interface

Aim: Create an interface having prototypes of functions area() and perimeter(). Create two classes Circle and Rectangle which implements the above interface. Create a menu driven program to find area and perimeter of objects.

Code:

import java.util.Scanner;  
interface obj {  
 double *pi*=3.14;  
 void area(double x,double y);  
 void peri(double x,double y);  
 void area(double x);  
 void peri(double x);  
}  
class rect implements obj{  
 @Override  
 public void area(double x, double y) {  
 System.*out*.print ("Area of rectangle is: "+(x\*y));  
 }  
 public void peri(double x,double y){  
 System.*out*.print ("Perimeter of rectangle is: "+(2\*(x+y)));  
 }

public void area(double x){ }  
 public void peri(double x){ }  
}  
class cir implements obj{  
 public void area(double x){  
 System.*out*.print ("Area of circle is: "+(*pi*\*x\*x));  
 }  
 public void peri(double x){  
 System.*out*.print ("Circumference of circle is: "+(2\**pi*\*x));  
 }  
 public void area(double x, double y) {}  
 public void peri(double x, double y) {}  
}  
class interfacearea{  
 public static void main(String arg[]){  
 rect r=new rect();  
 cir c=new cir();  
 int choice;  
 double l,b,s;  
 Scanner reader = new Scanner(System.*in*);  
 System.*out*.print("\nMENU \n1. Area of rectangle \n2.Area of circle \n3.Perimeter of rectangle \n4.Perimeter of circle \nEnter your choice:\n");  
 choice= reader.nextInt();  
 switch (choice)  
 {  
 case 1: System.*out*.print("Enter the length and breadth of rectangle: ");  
 l=reader.nextDouble();  
 b=reader.nextDouble();  
 r.area( l, b);  
 break;  
 case 2: System.*out*.print("Enter the radius of circle: ");  
 s=reader.nextDouble();  
 c.area(s);  
 break;  
 case 3: System.*out*.print("Enter the length and breadth of rectangle: ");  
 l=reader.nextDouble();  
 b=reader.nextDouble();  
 r.peri( l, b);  
 break;  
 case 4: System.*out*.print("Enter the radius of circle: ");  
 s=reader.nextDouble();  
 c.peri(s);  
 break;  
 default:System.*out*.print("Invalid choice");  
 }  
 }  
}

27.Interface Bill

Aim:

|  |
| --- |
| Prepare bill with the given format using calculate method from interface. |
| **Order No.** |
| **Date :** |
| **Product Id Name Quantity unit price Total** |
| **--------------------------------------------------------------------------------------------------------** |
| **101 A 2 25 50** |
| **102 B 1 100 100** |
| **--------------------------------------------------------------------------------------------------------** |
| **Net. Amount 150** |

Code:

import java.util.Date;  
import java.util.Scanner;  
interface bill {  
 float calcul();  
 void in();  
 void op();  
}  
class bills implements bill{  
 float price, net = 0;  
 int prod\_id,qty;  
 String name;  
 Scanner reader = new Scanner(System.*in*);  
 public void in(){  
 System.*out*.print("\nEnter product id: ");

prod\_id = reader.nextInt();  
 System.*out*.print("Enter product name: ");  
 name = reader.next();  
 System.*out*.print("Enter product quantity: ");  
 qty = reader.nextInt();  
 System.*out*.print("Enter price: ");  
 price = reader.nextFloat();  
 }  
 public void op(){  
 float tot=qty\*price;  
 System.*out*.print("\n" +prod\_id+" "+name+" "+qty+" "+price+" "+tot);  
 }  
 public float calcul(){  
 net=qty\*price;  
 return net;  
 }  
}  
class interfacebill{  
 public static void main(String args[]) {  
 Scanner reader = new Scanner(System.*in*);  
 int i,n,order\_no;  
 System.*out*.print("Enter order number: ");  
 order\_no = reader.nextInt();  
 Date date=new Date();  
 System.*out*.print(" Enter number of products: ");  
 n = reader.nextInt();  
 bills c = new bills();  
 bills b[] = new bills[n];  
 *// b.op(prod\_id,name,qty,price);  
 // b.calcul(qty,price);* for (i = 0; i < n; i++) {  
 b[i] = new bills();  
 b[i].in();  
 }  
 System.*out*.print(" \n\nORDER NUMBER: " +order\_no);  
 System.*out*.print(" \nDATE: " +date.toString());  
 System.*out*.print("\nPRODUCT ID NAME QUANTITY UNIT PRICE TOTAL");System.*out*.print("\n---------------------------------------------------------------------------");  
 for (i = 0; i < n; i++) {  
 b[i].op();  
 }  
 System.*out*.print("\n--------------------------------------------------------------------------");  
 float totalnet = 0;  
 for (i = 0; i < n; i++) {  
 b[i].calcul();  
 totalnet = totalnet + b[i].net;  
 }  
 if(i==n) {  
 System.*out*.print("\nNET AMOUNT: "+totalnet);  
 }  
 }  
}

28.Package Graphics

Aim: Create a Graphics package that has classes and interfaces for figures Rectangle, Triangle, Square and Circle. Test the package by finding the area of these figures.

Code:

package graphicss;  
import java.util.Scanner;  
 interface areafigure {  
 void tri();  
 void rect();  
 void sqr();  
 void cir();  
 }  
 public class fig{  
 Scanner reader=new Scanner(System.*in*);  
 void tri(){  
 System.*out*.print("\n Enter the breadth and height of triangle: ");  
 float b=reader.nextFloat();  
 float h=reader.nextFloat();  
 System.*out*.print("The area of triangle is: "+(0.5\*(b\*h)));  
 }  
 void rect(){  
 System.*out*.print("\n\n Enter the length and breadth of rectangle: ");  
 float l=reader.nextFloat();  
 float b=reader.nextFloat();  
 System.*out*.print("The area of rectangle is: "+(l\*b));  
 }  
 void sqr(){  
 System.*out*.print("\n\n Enter the side of square: ");  
 float s=reader.nextFloat();  
 System.*out*.print("The area of square is: "+(s\*s));  
 }  
 void cir() {  
 System.*out*.print("\n\n Enter the radius of circle: ");  
 float r = reader.nextFloat();  
 System.*out*.print("The area of circle is: " + (3.14 \* r \* r));  
 }  
}

package graphicss;  
public class figarea{  
 public static void main(String[] args) {  
 fig f = new fig();  
 f.tri();  
 f.rect();  
 f.sqr();  
 f.cir();  
 }  
}

29.Package Arithmetic

Aim: Create an Arithmetic package that has classes and interfaces for the 4 basic arithmetic operations. Test the package by implementing all operations on two given numbers

Code:

package arithmetic;  
import java.util.Scanner;  
public class operation {  
 public static void main(String[] args) {  
 Scanner reader = new Scanner(System.*in*);  
 System.*out*.print("Enter two numbers: ");  
 int a = reader.nextInt();  
 int b = reader.nextInt();  
 arthmtc m = new arthmtc();  
 m.add(a, b);  
 m.sub(a, b);  
 m.div(a, b);  
 m.multi(a, b);  
 }  
}

package arithmetic;  
interface arthop{  
 void add(int x,int y);  
 void sub(int x,int y);  
 void div(int x,int y);  
 void multi(int x,int y);  
}  
public class arthmtc {  
 void add(int x,int y){  
 System.*out*.println("ADDITION: "+(x+y));  
 }  
 void sub(int x,int y){  
 System.*out*.println("SUBTRACTION: "+(x-y));  
 }  
 void div(int x,int y){  
 System.*out*.println("DIVISION: "+(x/y));  
 }  
 void multi(int x,int y){  
 System.*out*.println("MULTIPLICATION: "+(x\*y));  
 }  
}

30.String Sort

Aim: Program to Sort strings

Code:

*//import java.util.Arrays;*import java.util.Scanner;  
public class sort {  
 public static void main(String args[]) {  
 int n,j;  
 String temp;  
 Scanner read = new Scanner(System.*in*);  
 System.*out*.print("Enter the size of array: ");  
 n = read.nextInt();  
 System.*out*.print("Enter values: ");  
 String[] array = new String[n];  
 for(int i = 0;i < n;i++)  
 {  
 array[i] = read.next();  
 }  
 for (int i = 0; i < n; i++) {  
 for ( j = i + 1; j < n; j++) {  
 if (array[i].compareTo(array[j]) > 0) {  
 temp = array[i];  
 array[i] = array[j];  
 array[j] = temp;  
 }  
 }  
 }  
 System.*out*.print("Strings in Sorted Order:");  
 for ( int i = 0; i <n ; i++)  
 {  
 System.*out*.print(array[i] + ", ");  
 }  
 }  
}

31.String Operations

Aim: Perform string manipulations (use switch statements and based on the choices given perform that operation)

Code:

import java.util.Scanner;  
public class stringop {  
 public static void main(String[] args) {  
 String s1, s2,s3,s4;  
 int ch,i,len,f=0;  
 Scanner reader = new Scanner(System.*in*);  
 System.*out*.print("\nSTRING OPERATIONS\n 1.Length\n2.Concat\n3.Compare\n4.Copy\nEnter your choice: ");  
 ch=reader.nextInt();  
 switch (ch){  
 case 1:System.*out*.print("Enter first string: ");  
 s1 = reader.next();  
 len=s1.length();  
 System.*out*.print("String length is: "+len);  
 break;  
 case 2:System.*out*.print("Enter first string: ");  
 s1 = reader.next();  
 System.*out*.print("Enter second string: ");  
 s2 = reader.next();  
 s4=s1.concat(s2);  
 s3=s1+s2;  
 System.*out*.print("Concatenated String: "+s3);  
 System.*out*.print("\nConcatination using library function: "+s4);  
 break;  
 case 3:System.*out*.print("Enter first string: ");  
 s1 = reader.next();  
 System.*out*.print("Enter second string: ");  
 s2 = reader.next();  
 */\*if(s1==s2)  
 f=1;  
 if(f==1)  
 System.out.print("\nStrings are not equal");  
 else {System.out.print("\nStrings are equal");  
 }\*/* System.*out*.print("\nCompare using library function: "+s1.equals(s2));  
 break;  
 case 4:System.*out*.print("Enter first string: ");  
 s1 = reader.next();  
 System.*out*.print("Enter second string: ");  
 s2 = reader.next();  
 s2=s1;  
 System.*out*.print("The copied string is : "+s2);  
 break;  
 default:System.*out*.print("Invalid choice");  
 break;  
 }  
 }  
}

32.Username and Password

Aim: Write a user defined exception class to authenticate the user name and password.

Code:

import java.util.Scanner;  
import java.lang.Exception;  
class UsernameException extends Exception {  
 UsernameException(String msg) {  
 System.*out*.print(msg);  
 }  
}  
class PasswordException extends Exception {  
 PasswordException(String msg) {  
 System.*out*.print(msg);  
 }  
}  
public class namepass {  
 public static void main(String[] args) {  
 Scanner reader = new Scanner(System.*in*);  
 String username, password;  
 System.*out*.print("Enter username: ");  
 username = reader.nextLine();  
 System.*out*.print("Enter password: ");  
 password = reader.nextLine();  
 int len = username.length();  
 try {  
 if(len < 8)  
 throw new UsernameException("Username must be greater than 8 characters\n");  
 else if(!password.equals("admin"))  
 throw new PasswordException("Incorrect password\n");  
 else  
 System.*out*.println("Login Successful");  
 }  
 catch (UsernameException u) {  
 }  
 catch (PasswordException p) {  
 }  
 }  
}

33.Average Exception

Aim: Find the average of N positive integers, raising a user defined exception for each negative input.

Code:

import java.util.Scanner;  
import java.lang.Exception;  
public class avg {  
 public static void main(String[] args){  
 try {  
 int i, n, sum=0;  
 double average=0;  
 int elem[]=new int[40];  
 Scanner reader = new Scanner(System.*in*);  
 System.*out*.print("Enter number of elements: ");  
 n = reader.nextInt();  
 System.*out*.println("Enter elements: ");  
 for (i = 0; i < n; i++) {  
 elem[i] = reader.nextInt();  
 if (elem[i] < 0){  
 throw new Exception("Number is negative");  
 }  
 *//System.out.print(elem[i]);* }  
 for(i=0;i<n;i++) {  
 if (elem[i] >= 0) {  
 sum = elem[i] + sum;  
 }  
 }average = sum/n;  
 System.*out*.print("\nAverage is " +average);  
 }  
 catch (Exception e) {  
 System.*out*.println(e);  
 }  
 }  
}

34. Thread

Aim: Define 2 classes; one for generating multiplication table of 5 and other for displaying first N prime numbers. Implement using threads. (Thread class)

Code:

import java.util.Scanner;  
 class tmultiple extends Thread  
 {  
 public void run()  
 {  
 System.*out*.println("Multiplication table of 5");  
 for(int i=1;i<=10;i++)  
 {  
 System.*out*.println(i+ " \* 5 = " +i\*5);  
 }  
 System.*out*.println();  
 }  
 }  
 class tprime extends Thread  
 {  
 public void run()  
 {  
 int i,j;  
 Scanner reader=new Scanner(System.*in*);  
 System.*out*.print("Enter the limit: ");  
 int n=reader.nextInt();  
 System.*out*.print("Prime numbers up to "+n+" are: ");  
 for(i=2;i<=n;i++)  
 {  
 int count=0;  
 for(j=1;j<=i;j++)  
 {  
 if(i%j==0)  
 count=count+1;  
 }  
 if(count==2)  
 System.*out*.print(i+" ");  
 }  
 }  
 }  
 class threadmulti  
 {  
 public static void main(String args[])  
 {  
 new tmultiple().start();  
 new tprime().start();  
 }  
 }

35.Thread Class

Aim: Define 2 classes; one for generating Fibonacci numbers and other for displaying even numbers in a given range. Implement using threads. (Runnable Interface)

Code:

import java.lang.Runnable;  
import java.lang.Thread;  
import java.util.Scanner;  
public class thread2 {  
 public static void main(String[] args) {  
 Runnable r1 = new Runnable1();  
 Thread t1 = new Thread(r1);  
 Runnable r2 = new Runnable2();  
 Thread t2 = new Thread(r2);  
 t1.start();  
 t2.start();  
 }  
}  
class Runnable1 implements Runnable{  
 public void run(){  
 int num1=0,num2=1,num3,i;  
 Scanner reader = new Scanner(System.*in*);  
 System.*out*.println("Enter the limit for fibonacci series :");  
 int n=reader.nextInt();  
 System.*out*.print("\nFibonacci series is: \n");  
 System.*out*.print(num1);  
 System.*out*.print( num2);  
 for(i=2;i<n;++i)  
 {  
 num3=num1+num2;  
 System.*out*.print(" "+num3);  
 num1=num2;  
 num2=num3;  
 }  
 }  
}  
class Runnable2 implements Runnable{  
 public void run(){  
 Scanner reader = new Scanner(System.*in*);  
 System.*out*.println("Enter the limit for even numbers :");  
 int n=reader.nextInt();  
 System.*out*.println("\nEven Numbers are :");  
 for(int i=1;i<=n;i++)  
 {  
 if(i%2==0){  
 System.*out*.println(i);  
 }  
 }  
 }  
}

36.Generic Stack

Aim: Program to create a generic stack and do the Push and Pop operations.

Code:

import java.util.\*;  
import java.io.\*;  
class stack<T> {  
 ArrayList<T> A;  
 int top = -1, size;  
 stack(int size) {  
 this.size = size;  
 this.A = new ArrayList<T>(size);  
 }  
 void push(T X) {  
 if (top + 1 == size) {  
 System.*out*.println("Stack Overflow");  
 } else {  
 top = top + 1;  
 if (A.size() > top)  
 A.set(top, X);  
 else  
 A.add(X);  
 }  
 }  
 T top() {  
 if (top == -1) {  
 System.*out*.println("Stack Underflow");  
 return null;  
 } else  
 return A.get(top);  
 }  
 void pop() {  
 if (top == -1) {  
 System.*out*.println("Stack Underflow");  
 } else  
 top--;  
 }  
 boolean empty() {  
 return top == -1;  
 }  
 public String toString() {  
 String Ans = "";  
 for (int i = 0; i < top; i++) {  
 Ans += String.*valueOf*(A.get(i)) + "->";  
 }  
 Ans += String.*valueOf*(A.get(top));  
 return Ans;  
 }  
}  
public class genericstack {  
 public static void main(String[] args)  
 {  
 stack<Integer> s1=new stack<>(3);  
 s1.push(10);  
 s1.push(20);  
 s1.push(30);  
 System.*out*.println("Integer stack s1 contains :\n" + s1);  
 s1.pop();  
 System.*out*.println("s1 after pop :\n" + s1);  
 stack<String> s2= new stack<>(3);  
 s2.push("Hello");  
 s2.push("World");  
 s2.push("Java");  
 System.*out*.println("\nString stack s2 contents: \n" + s2);  
 stack<Float> s3= new stack<>(2);  
 s3.push(100.0f);  
 s3.push(200.0f);  
 System.*out*.println("\nFloat stack s3 contents: \n" + s3);  
 System.*out*.println("\nTop element of s3 : \n" + s3.top());  
 }  
}

37.Bubble Sort

Aim: Using generic method perform Bubble sort.

Code:

import java.util.Arrays;  
 public class Bubblesort<T>  
 {  
 T[] array;  
 public static<T extends Comparable> void sort(T[] array)  
 {  
 for(int i = array.length; i > 1; i--)  
 {  
 for(int j = 0; j < i - 1; j++)  
 {  
 if(array[j].compareTo(array[j+1]) > 0)  
 {  
 T temp = array[j];  
 array[j] = array[j+1];  
 array[j+1] = temp;  
 }  
 }  
 }  
 System.*out*.println("Sorted array- " + Arrays.*toString*(array));  
 *//System.out.println("Sorted array- " + array);* }  
 public static void main(String[] args)  
 { Integer[] arr = {20,100,30,10,80,66};  
 Bubblesort<Integer> obj = new Bubblesort<Integer>();  
 obj.*sort*(arr);  
 String[] arr2 = {"Siyah","Athira","Minion","Anjali","Revu","Archana","Sreejith"};  
 Bubblesort<String> obj2 = new Bubblesort<String>();  
 obj2.*sort*(arr2);  
 }  
 }

38.Sub Directories in file

Aim: Program to list the sub directories and files in a given directory and also search for a file name.

Code:

import java.io.File;  
public class files {  
 static void recursivePrint(File[] arr,int index,int level){  
 if(index==arr.length)  
 return;  
 for(int i=0;i<level;i++)  
 System.*out*.print("\t");  
 if(arr[index].isFile())  
 System.*out*.println(arr[index].getName());  
 else if(arr[index].isDirectory())  
 {  
 System.*out*.println("[" +arr[index].getName() + "]");  
 *recursivePrint*(arr[index].listFiles(),0,level+1);  
 }  
 *recursivePrint*(arr,++index,level);  
 }  
 public static void main(String[] args){  
 String maindirpath="C:\\Users\\ATHIRA\\IdeaProjects\\AthiraProject1\\src\\arithmetic";  
 File maindir=new File(maindirpath);  
 if(maindir.exists() && maindir.isDirectory()){  
 File arr[]=maindir.listFiles();  
 System.*out*.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");  
 System.*out*.println("Files from main directory :" +maindir);  
 System.*out*.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");  
 *recursivePrint*(arr,0,0);  
 }  
 }  
}

39.Read and Write File

Aim: Write a program to write to a file, then read from the file and display the contents on the console.

Code:

import java.io.FileReader;  
import java.io.FileWriter;  
import java.io.IOException;  
class fileio {  
 public static void main(String[] args){  
 try{  
 FileReader fr=new FileReader("C:\\Users\\ATHIRA\\IdeaProjects\\AthiraProject1\\src\\input.txt");  
 FileWriter fw=new FileWriter("C:\\Users\\ATHIRA\\IdeaProjects\\AthiraProject1\\src\\output.txt");  
 String str="";  
 int i;  
 while((i=fr.read())!=-1){  
 str+=(char)i;  
 }  
 System.*out*.println(str);  
 fw.write(str);  
 fr.close();  
 fw.close();  
 System.*out*.println("File reading and writing both done");  
 }  
 catch (IOException e){  
 System.*out*.println("There are some IOException");  
 }  
 }  
}

40.Copy File

Aim: Write a program to copy one file to another.

Code:

import java.io.\*;  
public class CopyFile  
{  
 public static void main(String args[])throws IOException,FileNotFoundException  
 {  
 int i;  
 File f1=new File("C:\\Users\\ATHIRA\\IdeaProjects\\AthiraProject1\\src\\input.txt");  
 File f2=new File("C:\\Users\\ATHIRA\\IdeaProjects\\AthiraProject1\\src\\hello.txt");  
 FileInputStream in=new FileInputStream(f1);  
 FileOutputStream op=new FileOutputStream(f2);  
 do  
 {  
 i=in.read();  
 if(i!=-1)  
 op.write(i);  
 }  
 while(i!=-1);  
 in.close();  
 op.close();  
 }  
}

41.Odd Even File

Aim: Write a program that reads from a file having integers. Copy even numbers and odd numbers to separate files.

Code:

import java.io.\*;  
public class OddEven  
{  
 public static void main(String args[])throws IOException,FileNotFoundException  
 {  
 int i;  
 File f1=new File("C:\\Users\\ATHIRA\\IdeaProjects\\AthiraProject1\\src\\OddEven.txt ");  
 File f2=new File("C:\\Users\\ATHIRA\\IdeaProjects\\AthiraProject1\\src\\Even.txt ");  
 File f3=new File("C:\\Users\\ATHIRA\\IdeaProjects\\AthiraProject1\\src\\Odd.txt ");  
 FileInputStream in=new FileInputStream(f1);  
 FileOutputStream op1=new FileOutputStream(f2);  
 FileOutputStream op2=new FileOutputStream(f3);  
 do  
 {  
 i=in.read();  
 if(i%2==0)  
 op1.write(i);  
 else  
 op2.write(i);  
 }  
 while(i!=-1);  
 in.close();  
 op1.close();  
 op2.close();  
 }  
}

42.Linked Stack

Aim: Program to create Stack and do the push and pop operations.

Code:

import static java.lang.System.*exit*;  
class StackUsingLinkedlist {  
 private class Node {  
 int data;  
 Node link;  
 }  
 Node top;  
 StackUsingLinkedlist()  
 {  
 this.top = null;  
 }  
 public void push(int x)  
 {  
 Node temp = new Node();  
 if (temp == null) {  
 System.*out*.print("\nHeap Overflow");  
 return;  
 }  
 temp.data = x;  
 temp.link = top;  
 top = temp;  
 }  
 public boolean isEmpty()  
 {  
 return top == null;  
 }  
 public int peek()  
 {  
 if (!isEmpty()) {  
 return top.data;  
 }  
 else {  
 System.*out*.println("Stack is empty");  
 return -1;  
 }  
 }  
 public void pop()  
 {  
 if (top == null) {  
 System.*out*.print("\nStack Underflow");  
 return;  
 }  
 top = (top).link;  
 }  
 public void display()  
 {  
 if (top == null) {  
 System.*out*.printf("\nStack Underflow");  
 *exit*(1);  
 }  
 else {  
 Node temp = top;  
 while (temp != null) {  
 System.*out*.printf("%d->", temp.data);  
 temp = temp.link;  
 }  
 }  
 }  
}  
public class StackLinkedLIst {  
 public static void main(String[] args)  
 {  
 StackUsingLinkedlist obj = new StackUsingLinkedlist();  
 obj.push(11);  
 obj.push(22);  
 obj.push(33);  
 obj.push(44);  
 obj.display();  
 System.*out*.printf("\nTop element is %d\n", obj.peek());  
 obj.pop();  
 obj.pop();  
 obj.display();  
 System.*out*.printf("\nTop element is %d\n", obj.peek());  
 }  
}

43.Linked List Remove Elements

Aim: Program to remove all the elements from a linked list

Code:

import java.util.\*;  
import java.io.\*;  
 public class removefiles  
 {  
 public static void main(String[] args) {  
 LinkedList<String> l\_list = new LinkedList<String>();  
 l\_list.add("A");  
 l\_list.add("B");  
 l\_list.add("C");  
 l\_list.add("D");  
 l\_list.add("E");  
 l\_list.add("F");  
 System.*out*.println("The Original linked list: " + l\_list);  
 l\_list.clear();  
 System.*out*.println("The New linked list: " + l\_list);  
 }  
 }

44.Clinet UDP

Aim: Client Server communication using DatagramSocket - UDP

Code:

import java.io.\*;  
import java.net.\*;  
import java.util.\*;  
public class UDPclient  
{  
 DatagramSocket ds;  
 DatagramPacket dp,dp1;  
 byte[] sendPacket;  
 byte[] receivePacket;  
 BufferedReader in;  
 InetAddress ip;  
 int Port;  
 String str;  
 UDPclient()  
 {  
 try  
 {  
 in=new BufferedReader(new InputStreamReader(System.*in*));  
 ip=InetAddress.*getByName*("localhost");  
 ds=new DatagramSocket();  
 while(true)  
 {  
 sendPacket=new byte[100];  
 receivePacket=new byte[100];  
 dp=new DatagramPacket(receivePacket,receivePacket.length);  
 System.*out*.print("Client:");  
 str=in.readLine();  
 sendPacket=str.getBytes();  
 dp1=new DatagramPacket(sendPacket,sendPacket.length,ip,1456);  
 ds.send(dp1);  
 if(str.equals("exit"))  
 System.*exit*(0);  
 ds.receive(dp);  
 String data=new String(dp.getData());  
 System.*out*.println("Server:"+data);  
 }  
 }catch(Exception e){}  
 }  
 public static void main(String args[])  
 {  
 UDPclient s=new UDPclient();  
 }  
}

45.Server UDP

Code:

import java.io.\*;  
import java.net.\*;  
import java.util.\*;  
public class UDPserver  
{  
 DatagramSocket ds;  
 DatagramPacket dp,dp1;  
 byte[] sendPacket;  
 byte[] receivePacket;  
 BufferedReader in;  
 InetAddress ip;  
 int Port;  
 String str;  
 UDPserver()  
 {  
 try  
 {  
 in=new BufferedReader(new InputStreamReader(System.*in*));  
 ds=new DatagramSocket(1456);  
 sendPacket=new byte[100];  
 receivePacket=new byte[100];  
 while(true)  
 {  
 sendPacket=new byte[100];  
 receivePacket=new byte[100];  
 dp=new DatagramPacket(receivePacket,receivePacket.length);  
 ds.receive(dp);  
 String data=new String(dp.getData());  
 System.*out*.println("Client:"+data);  
 System.*out*.print("Server: ");  
 str=in.readLine();  
 ip=dp.getAddress();  
 Port=dp.getPort();  
 sendPacket=str.getBytes();  
 dp1=new DatagramPacket(sendPacket,sendPacket.length,ip,Port);  
 ds.send(dp1);  
 if(str.equals("exit"))  
 System.*exit*(0);  
 }  
 }catch(Exception e){}  
 }  
 public static void main(String args[])  
 {  
 UDPserver s=new UDPserver();  
 }  
}

46.Client TCP

Aim: Client Server communication using Socket - TCP/IP

Code:

import java.io.\*;  
import java.net.\*;  
 public class TCPclient  
 {  
 public static void main(String[] args) throws IOException  
 {  
 System.*out*.println("TCP CLIENT");  
 System.*out*.println("Enter the host name to connect");  
 DataInputStream inp=new DataInputStream(System.*in*);  
 String str=inp.readLine();  
 Socket clientsoc=new Socket(str,9);  
 PrintWriter out=new PrintWriter(clientsoc.getOutputStream(),true);  
 BufferedReader in=new BufferedReader(new InputStreamReader(clientsoc.getInputStream()));  
 BufferedReader stdin=new BufferedReader(new InputStreamReader(System.*in*));  
 String userinput;  
 try  
 {  
 while(true)  
 {  
 System.*out*.println("Server says: "+ in.readLine());  
 userinput=stdin.readLine();  
 out.println(userinput);  
 }  
 }  
 catch(Exception e)  
 {  
 System.*exit*(0);  
 }  
 }  
 }

47.Server TCP

Code:

import java.io.\*;  
import java.net.\*;  
public class TCPserver  
{  
 public static void main(String a[]) throws Exception  
 {  
 System.*out*.println("TCP SERVER");  
 System.*out*.println("Server is ready to connect");  
 ServerSocket serversoc=new ServerSocket(9);  
 Socket clientsoc=serversoc.accept();  
 PrintWriter out=new PrintWriter(clientsoc.getOutputStream(),true);  
 BufferedReader in=new BufferedReader(new InputStreamReader(clientsoc.getInputStream()));  
 String inputline;  
 BufferedReader stdin=new BufferedReader(new InputStreamReader(System.*in*));  
 try  
 {  
 while(true)  
 {  
 inputline=stdin.readLine();  
 out.println(inputline);  
 System.*out*.println("Client says: "+ in.readLine());  
 }  
 }  
 catch(Exception e)  
 {  
 System.*exit*(0);  
 }  
 }  
}

48. Queue Linked List

Aim:

Code:

class QNode {  
 int key;  
 QNode next;  
 public QNode(int key)  
 {  
 this.key = key;  
 this.next = null;  
 }  
}  
class Queue {  
 QNode front, rear;  
 public Queue()  
 {  
 this.front = this.rear = null;  
 }  
 void enqueue(int key)  
 {  
 QNode temp = new QNode(key);  
 if (this.rear == null) {  
 this.front = this.rear = temp;  
 return;  
 }  
 this.rear.next = temp;  
 this.rear = temp;  
 }  
 void dequeue()  
 {  
 if (this.front == null)  
 return;  
 QNode temp = this.front;  
 this.front = this.front.next;  
 if (this.front == null)  
 this.rear = null;  
 }  
}  
public class QueueLinkedList {  
 public static void main(String[] args)  
 {  
 Queue q = new Queue();  
 q.enqueue(10);  
 q.enqueue(20);  
 q.dequeue();  
 q.dequeue();  
 q.enqueue(30);  
 q.enqueue(40);  
 q.enqueue(50);  
 q.dequeue();  
 System.*out*.println("Queue Front : " + q.front.key);  
 System.*out*.println("Queue Rear : " + q.rear.key);  
 }  
}

49.

Aim:

Code: