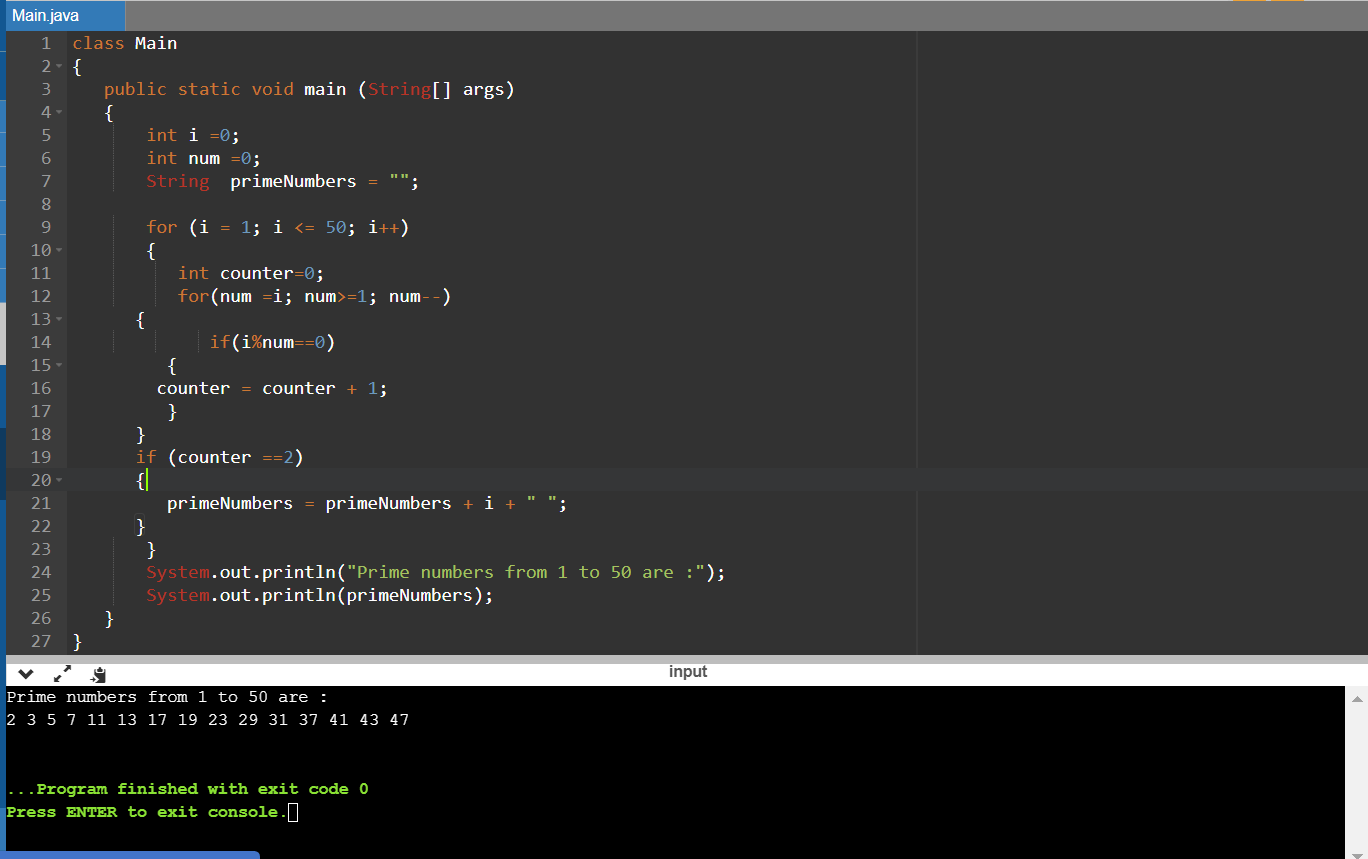
21/12/2020

1 Write a program in Java to print all prime numbers from 1 to 50



class Main

{

public static void main (String[] args)

{

int i =0;

int num =0;

String primeNumbers = "";

for (i = 1; i <= 50; i++)

{

int counter=0;

for(num =i; num>=1; num--)

{

if(i%num==0)

{

counter = counter + 1;

}

}

if (counter ==2)

{

primeNumbers = primeNumbers + i + " ";

}

}

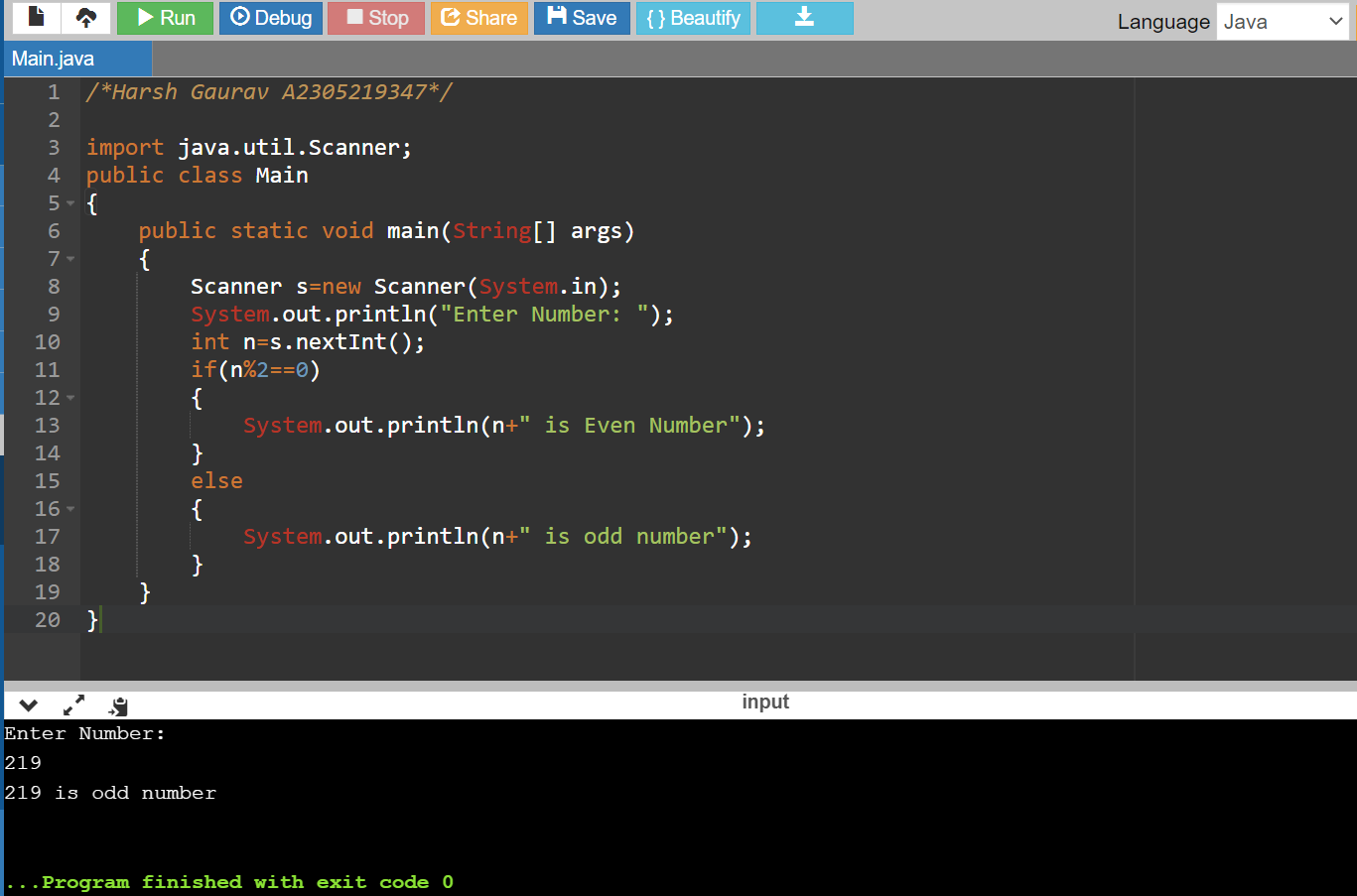
System.out.println("Prime numbers from 1 to 50 are :");

System.out.println(primeNumbers);

}

}

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



/\*Harsh Gaurav A2305219347\*/

import java.util.Scanner;

public class Main

{

public static void main(String[] args)

{

Scanner s=new Scanner(System.in);

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

int n=s.nextInt();

if(n%2==0)

{

System.out.println(n+" is Even Number");

}

else

{

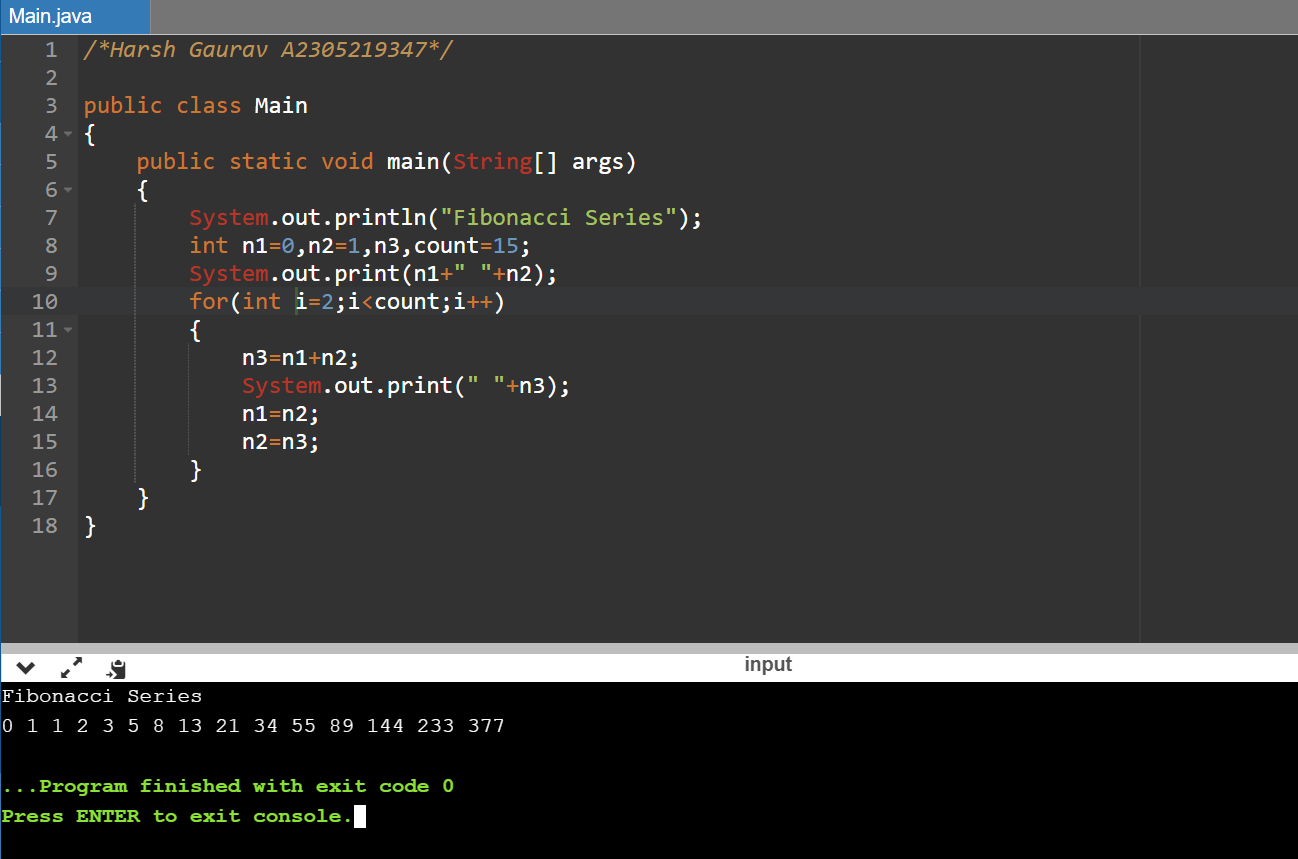
System.out.println(n+" is odd number");

}

}

}

3 .Write a program to print a Fibonacci series upto a limit.



/\*Harsh Gaurav A2305219347\*/

public class Main

{

public static void main(String[] args)

{

System.out.println("Fibonacci Series");

int n1=0,n2=1,n3,count=15;

System.out.print(n1+" "+n2);

for(int i=2;i<count;i++)

{

n3=n1+n2;

System.out.print(" "+n3);

n1=n2;

n2=n3;

}

}

}

4. Write a program to check whether a year is leap year.



/\*Harsh Gaurav A2305219347\*/

import java.util.Scanner;

public class Main

{

public static void main(String[] args)

{

Scanner s=new Scanner(System.in);

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

int n=s.nextInt();

if(n%4==0&&n%100!=0)

{

System.out.println(n+" is Leap Year");

}

else if(n%400==0)

{

System.out.println(n+" is Leap Year");

}

else

{

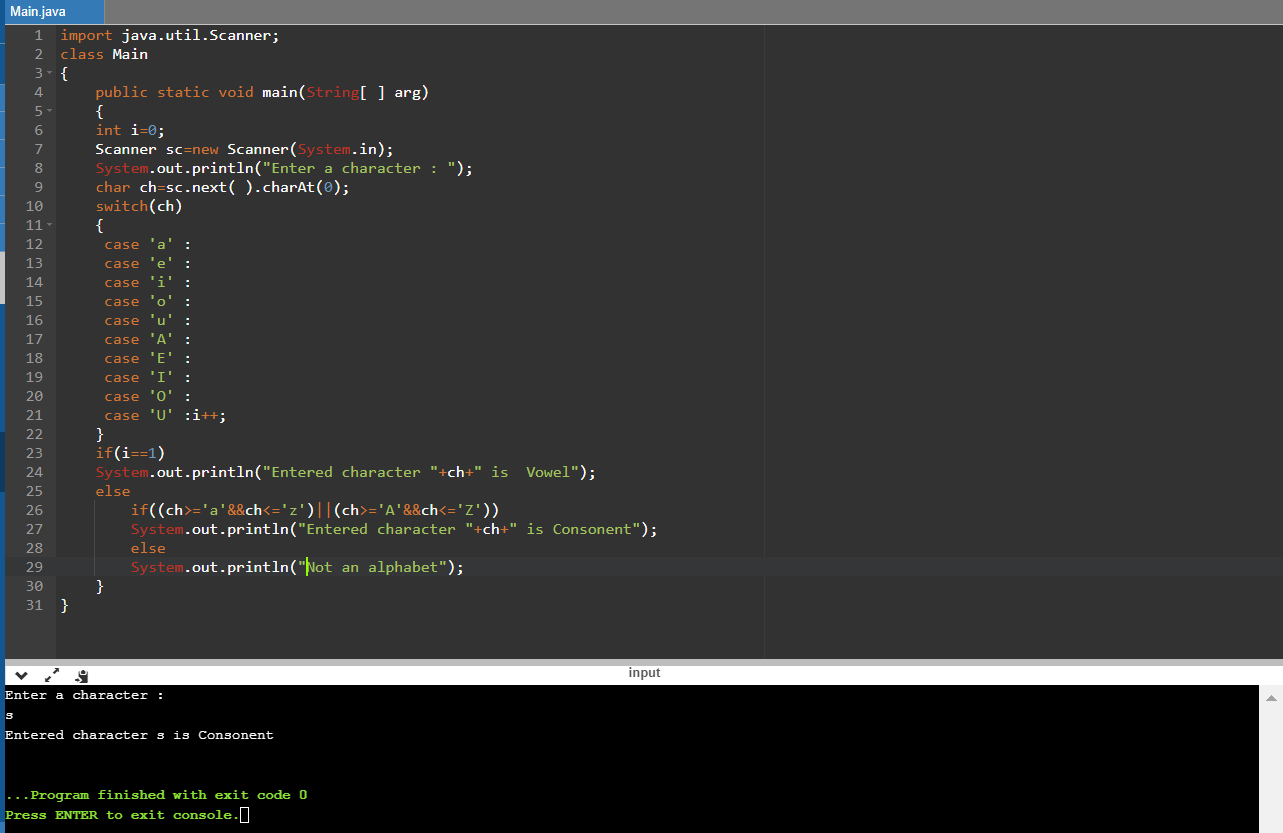
System.out.println(n+"is not a Leap Year ");

}

}

}

5. Write a program to check if an input character is vowel or constant; if it is none, display error. Write a program to check if an input character is vowel or constant; if it is none, display error.



import java.util.Scanner;

class Main

{

public static void main(String[ ] arg)

{

int i=0;

Scanner sc=new Scanner(System.in);

System.out.println("Enter a character : ");

char ch=sc.next( ).charAt(0);

switch(ch)

{

case 'a' :

case 'e' :

case 'i' :

case 'o' :

case 'u' :

case 'A' :

case 'E' :

case 'I' :

case 'O' :

case 'U' :i++;

}

if(i==1)

System.out.println("Entered character "+ch+" is Vowel");

else

if((ch>='a'&&ch<='z')||(ch>='A'&&ch<='Z'))

System.out.println("Entered character "+ch+" is Consonent");

else

System.out.println("Not an alphabet");

}

}

6. Write a program to check whether a number is prime



import java.util.Scanner;

public class Main{

public static void main(String args[]){

int i,m=0,flag=0;

Scanner s=new Scanner(System.in);

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

int n=s.nextInt();

m=n/2;

if(n==0||n==1){

System.out.println(n+" is not prime number");

}else{

for(i=2;i<=m;i++){

if(n%i==0){

System.out.println(n+" is not prime number");

flag=1;

break;

}

}

if(flag==0) { System.out.println(n+" is prime number"); }

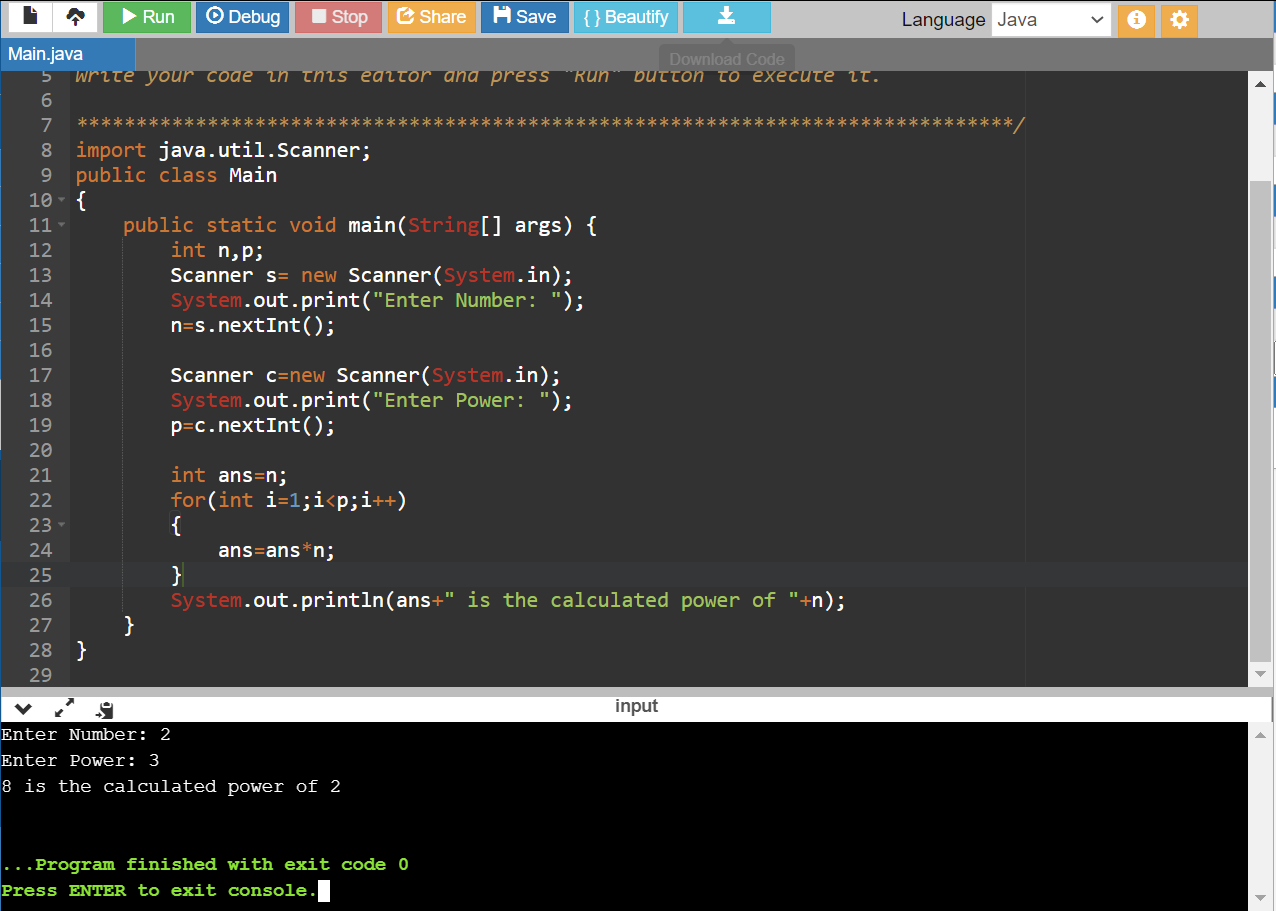
}

}

}

4/1/2021

7. Write a program to calculate power of a number. (without Math.pow)



import java.util.Scanner;

public class Main

{

public static void main(String[] args) {

int n,p;

Scanner s= new Scanner(System.in);

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

n=s.nextInt();

Scanner c=new Scanner(System.in);

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

p=c.nextInt();

int ans=n;

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

{

ans=ans\*n;

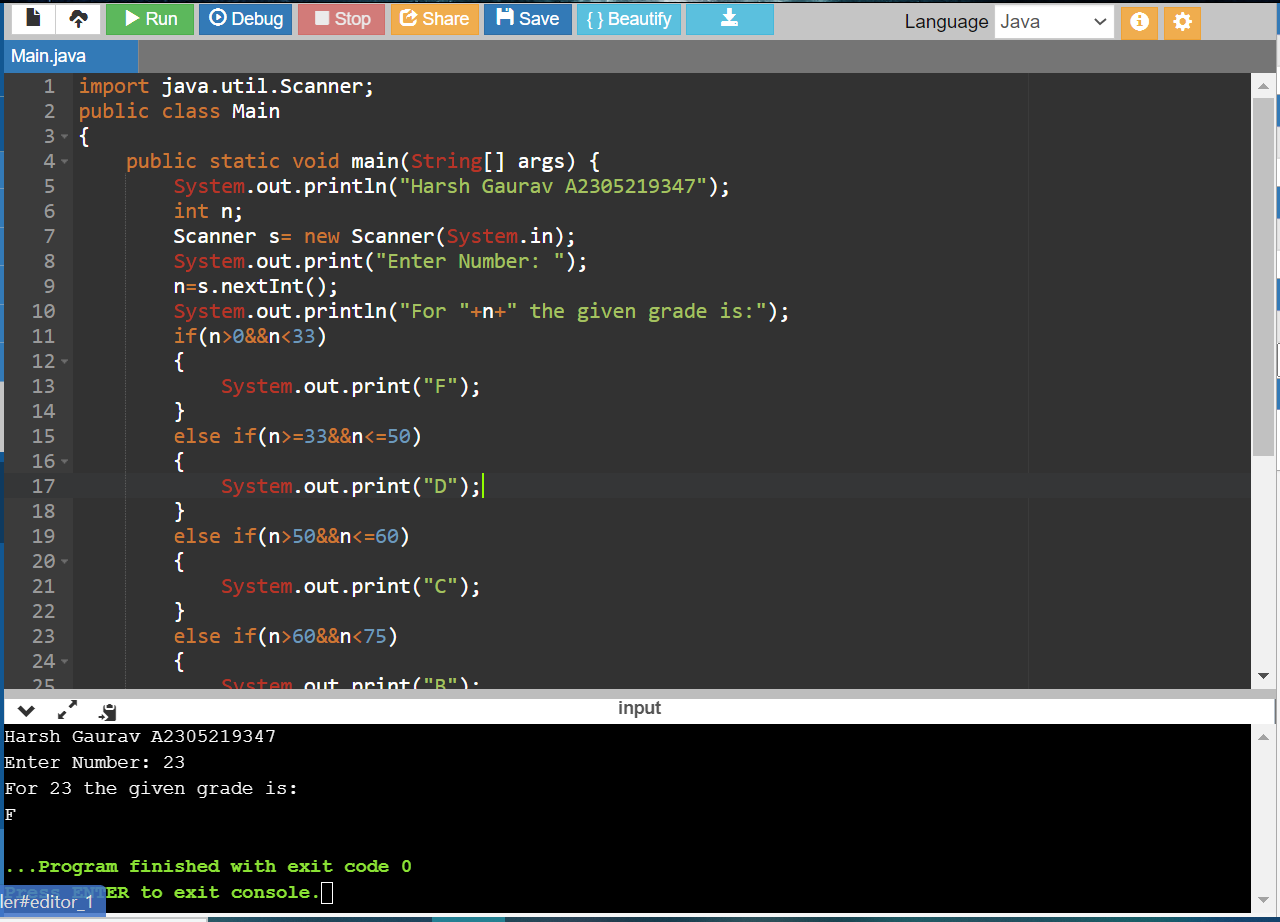
}

System.out.println(ans+" is the calculated power of "+n);

}

}

8. Write a program to display grade of students.



import java.util.Scanner;

public class Main

{

public static void main(String[] args) {

System.out.println("Harsh Gaurav A2305219347");

int n;

Scanner s= new Scanner(System.in);

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

n=s.nextInt();

System.out.println("For "+n+" the given grade is:");

if(n>0&&n<33)

{

System.out.print("F");

}

else if(n>=33&&n<=50)

{

System.out.print("D");

}

else if(n>50&&n<=60)

{

System.out.print("C");

}

else if(n>60&&n<75)

{

System.out.print("B");

}

else if(n>=75&&n<=100)

{

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

}

else

{

System.out.print("Not Valid");

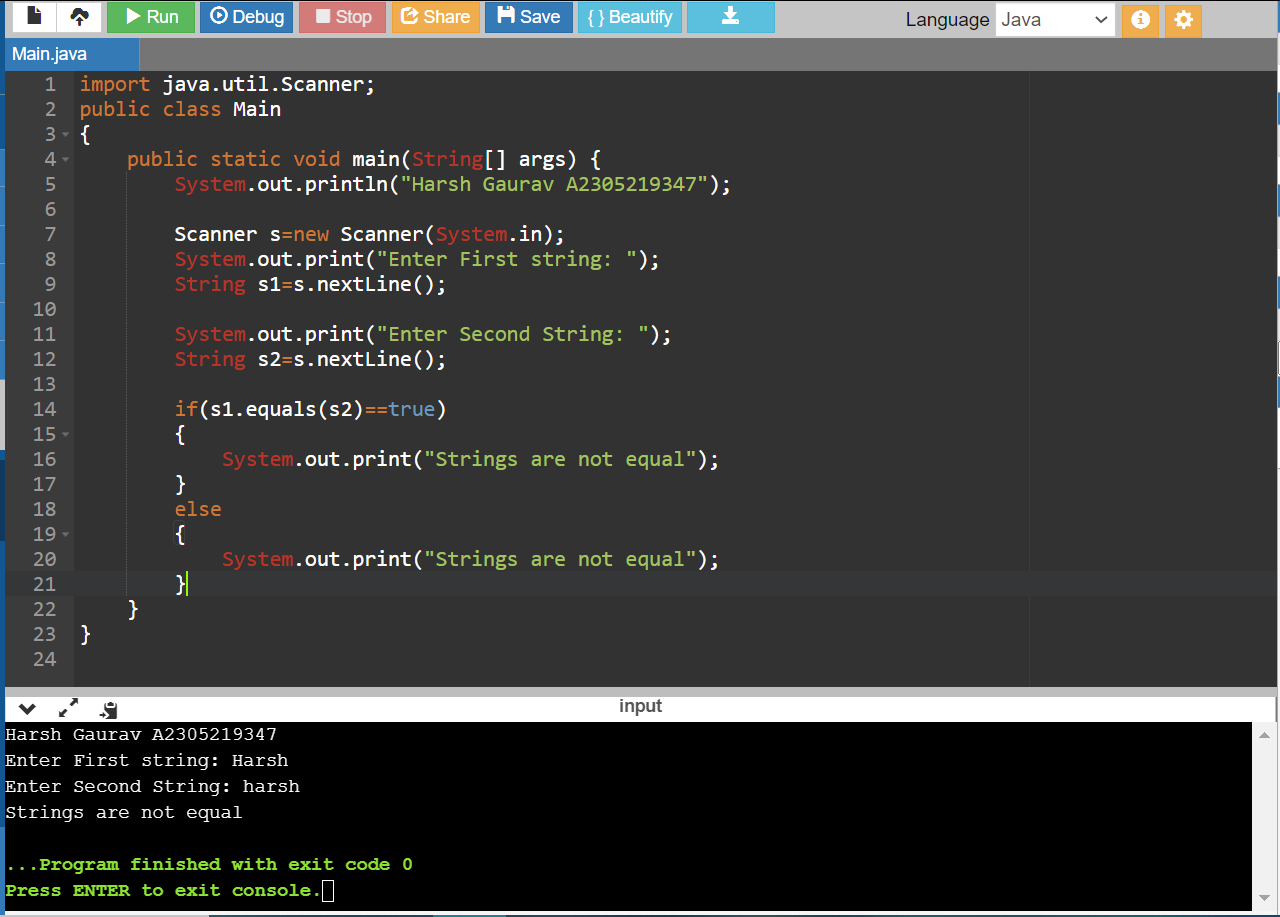
return;

}

}

}

9. Write a program to check whether two strings are equal or not.



import java.util.Scanner;

public class Main

{

public static void main(String[] args) {

System.out.println("Harsh Gaurav A2305219347");

Scanner s=new Scanner(System.in);

System.out.print("Enter First string: ");

String s1=s.nextLine();

System.out.print("Enter Second String: ");

String s2=s.nextLine();

if(s1.equals(s2)==true)

{

System.out.print("Strings are not equal");

}

else

{

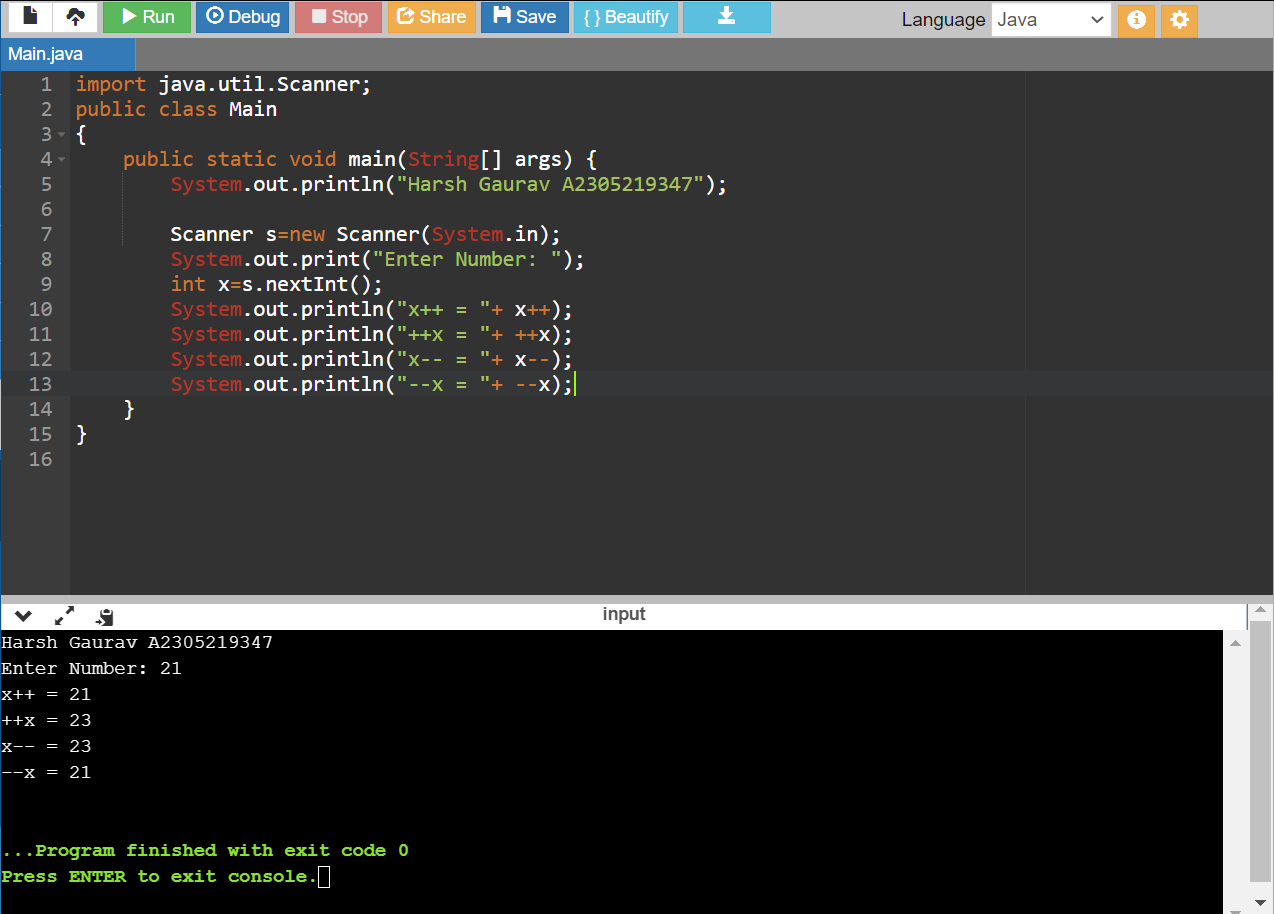
System.out.print("Strings are not equal");

}

}

}

10. Write a program to display numbers using increment and decrement operators.



import java.util.Scanner;

public class Main

{

public static void main(String[] args) {

System.out.println("Harsh Gaurav A2305219347");

Scanner s=new Scanner(System.in);

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

int x=s.nextInt();

System.out.println("x++ = "+ x++);

System.out.println("++x = "+ ++x);

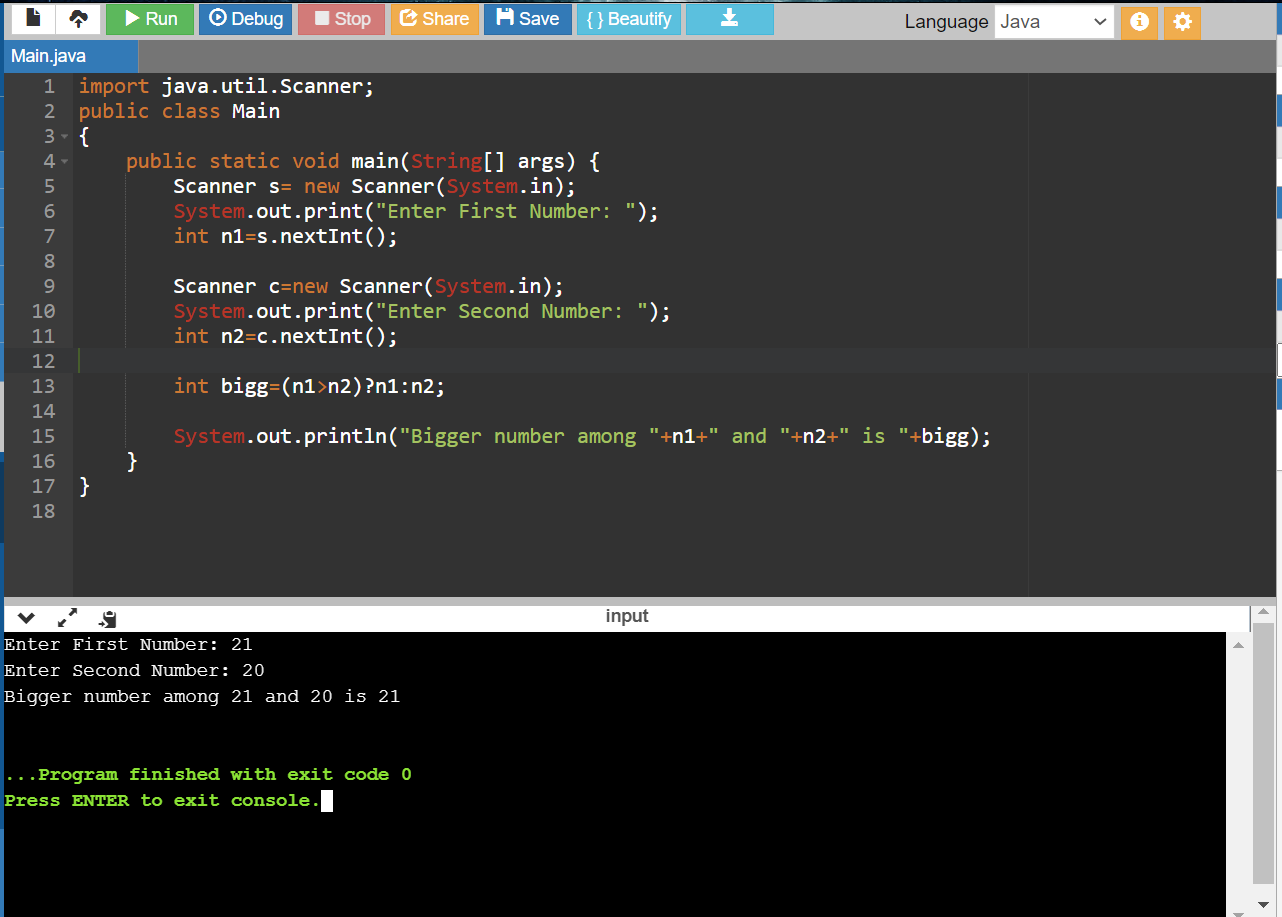
System.out.println("x-- = "+ x--);

System.out.println("--x = "+ --x);

}

}

11. Write a program to find greater number out of two using ternary operator.



import java.util.Scanner;

public class Main

{

public static void main(String[] args) {

Scanner s= new Scanner(System.in);

System.out.print("Enter First Number: ");

int n1=s.nextInt();

Scanner c=new Scanner(System.in);

System.out.print("Enter Second Number: ");

int n2=c.nextInt();

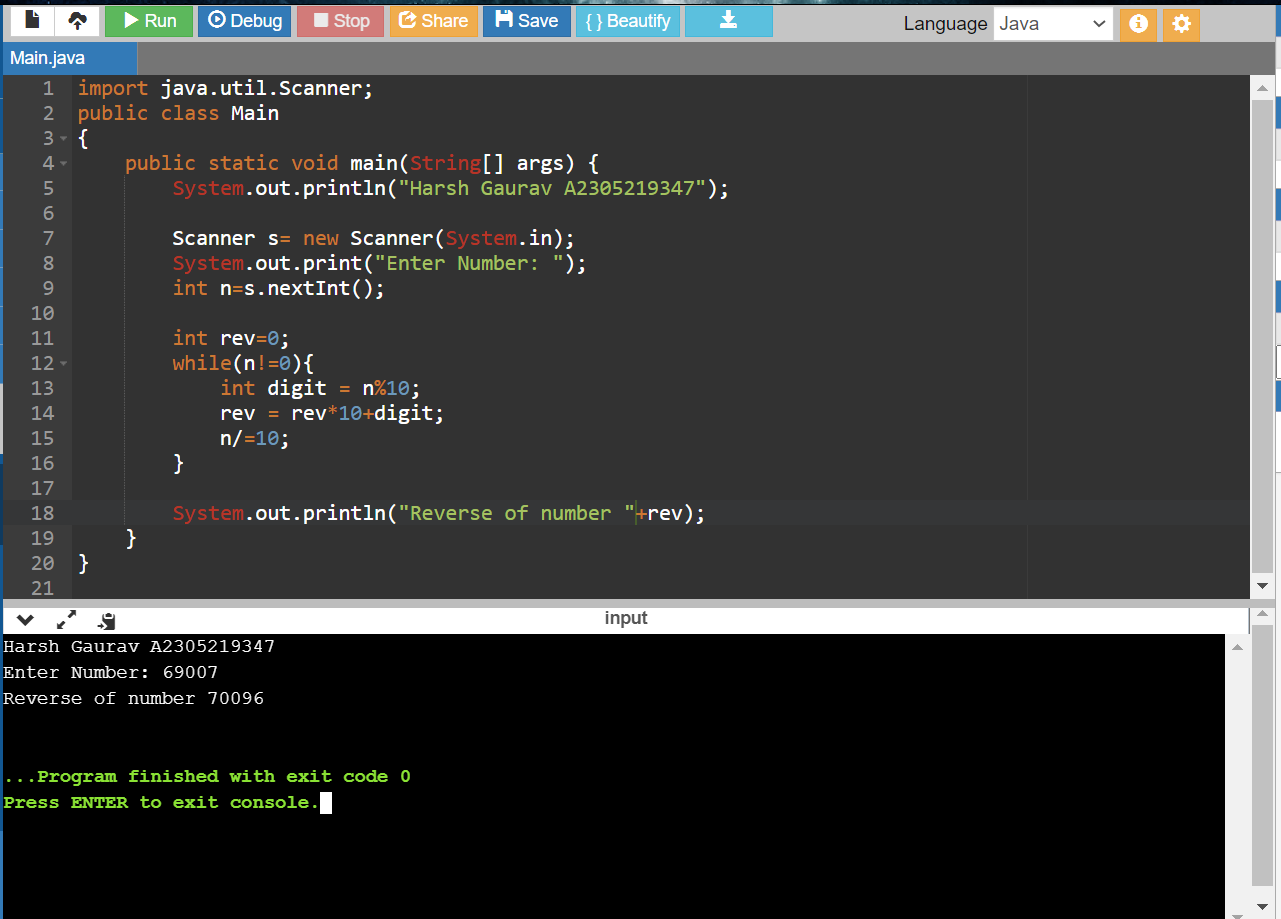
int bigg=(n1>n2)?n1:n2;

System.out.println("Bigger number among "+n1+" and "+n2+" is "+bigg);

}

}

12. Write a program to reverse a given number.



import java.util.Scanner;

public class Main

{

public static void main(String[] args) {

System.out.println("Harsh Gaurav A2305219347");

Scanner s= new Scanner(System.in);

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

int n=s.nextInt();

int rev=0;

while(n!=0){

int digit = n%10;

rev = rev\*10+digit;

n/=10;

}

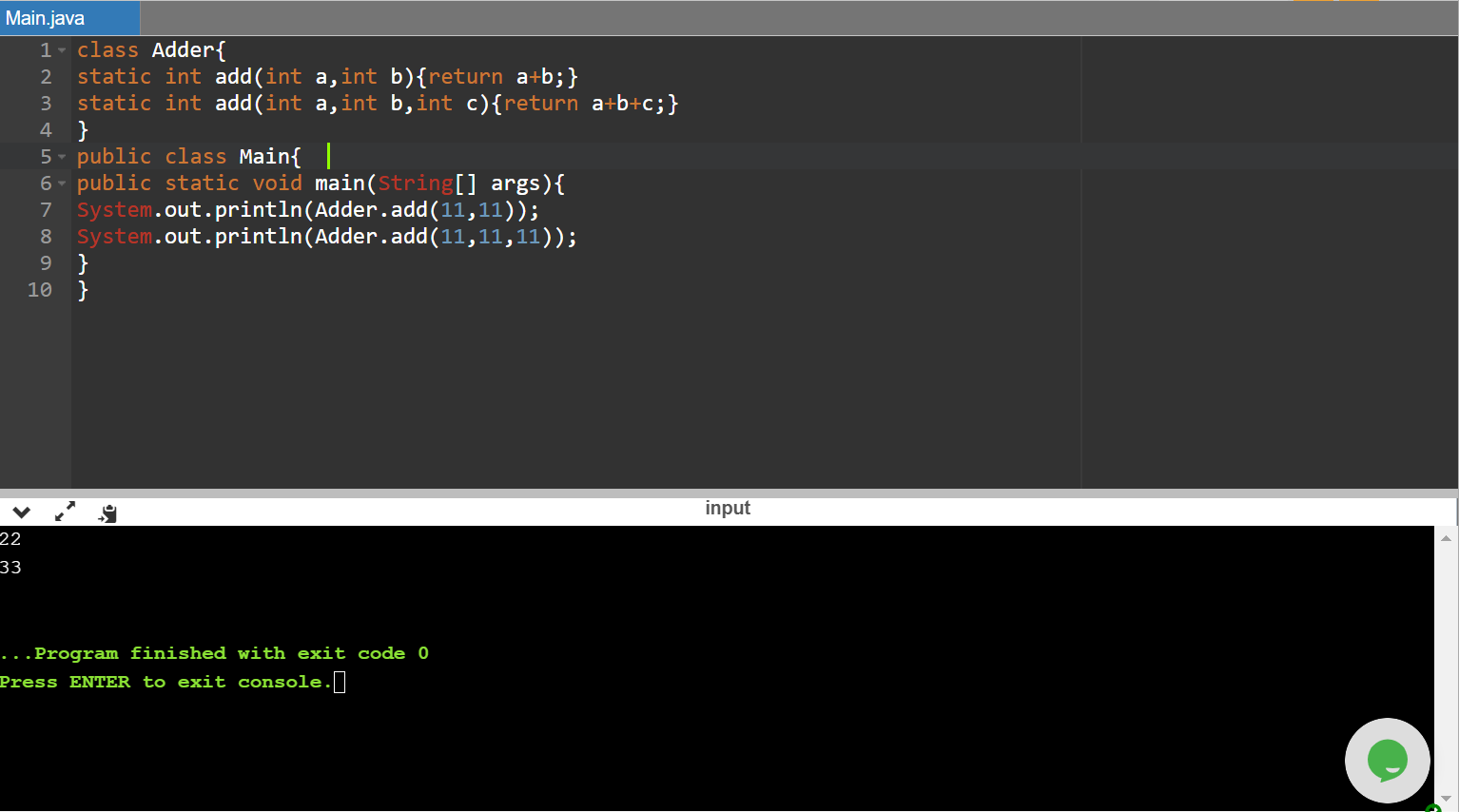
System.out.println("Reverse of number "+rev);

}

}

15/1/2020

Q13. Write a program to show method overloading.



class Adder{

static int add(int a,int b){return a+b;}

static int add(int a,int b,int c){return a+b+c;}

}

public class Main{

public static void main(String[] args){

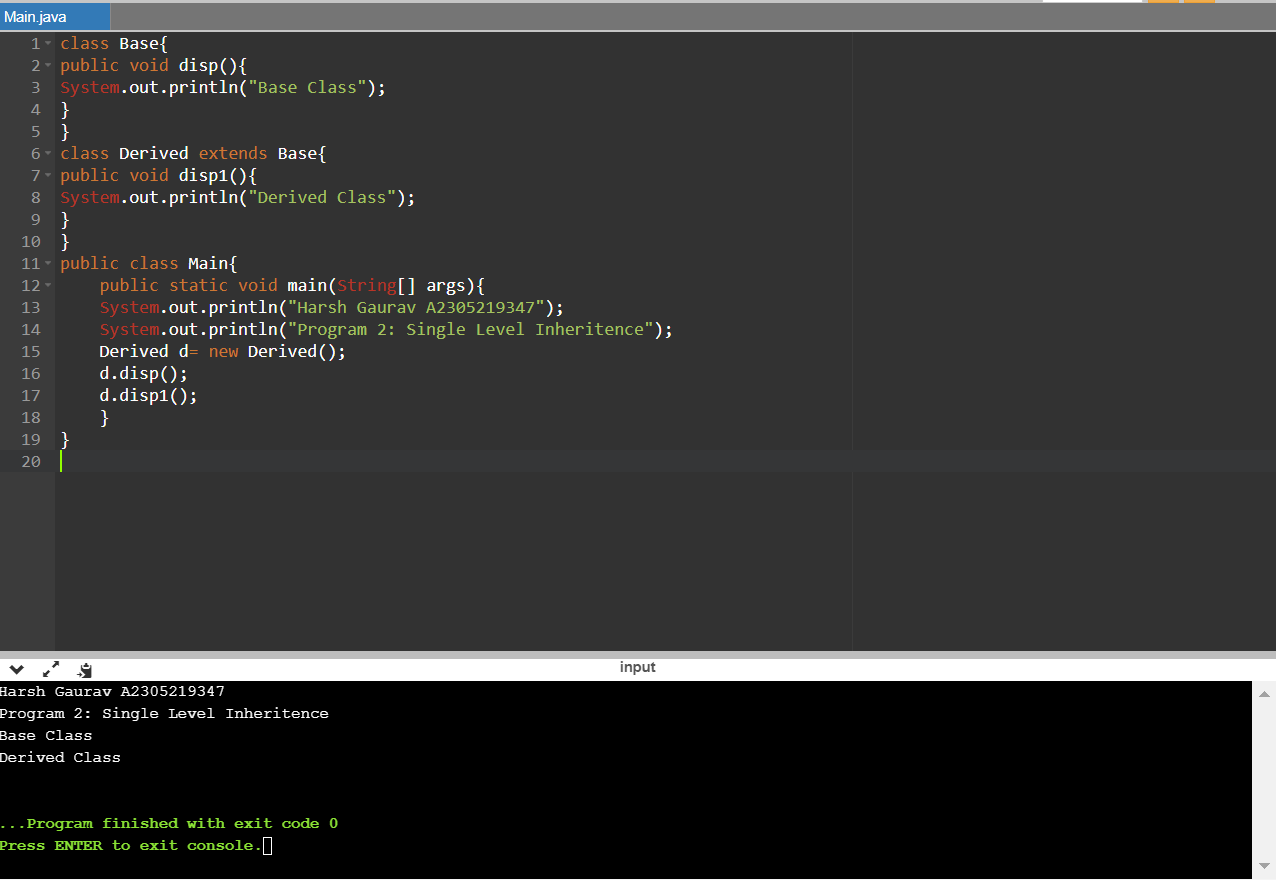
System.out.println(Adder.add(11,11));

System.out.println(Adder.add(11,11,11));

}

}

Q14. Write a program to show single level inheritance.



class Base{

public void disp(){

System.out.println("Base Class");

}

}

class Derived extends Base{

public void disp1(){

System.out.println("Derived Class");

}

}

public class Main{

public static void main(String[] args){

System.out.println("Harsh Gaurav A2305219347");

System.out.println("Program 14: Single Level Inheritence");

Derived d= new Derived();

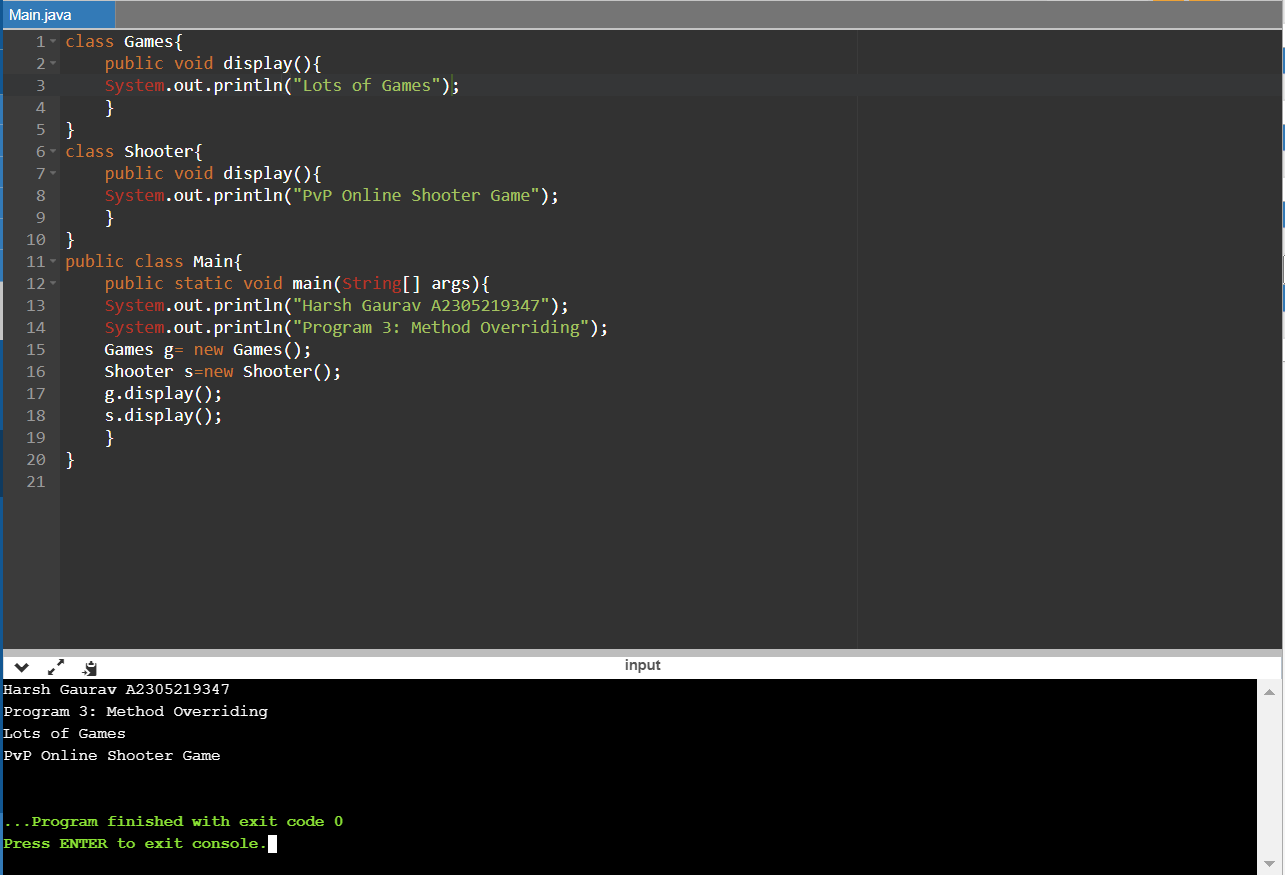
d.disp();

d.disp1();

}

}

Q15. Write a program to show method overriding.



class Games{

public void display(){

System.out.println("Lots of Games");

}

}

class Shooter{

public void display(){

System.out.println("PvP Online Shooter Game");

}

}

public class Main{

public static void main(String[] args){

System.out.println("Harsh Gaurav A2305219347");

System.out.println("Program 15: Method Overriding");

Games g= new Games();

Shooter s=new Shooter();

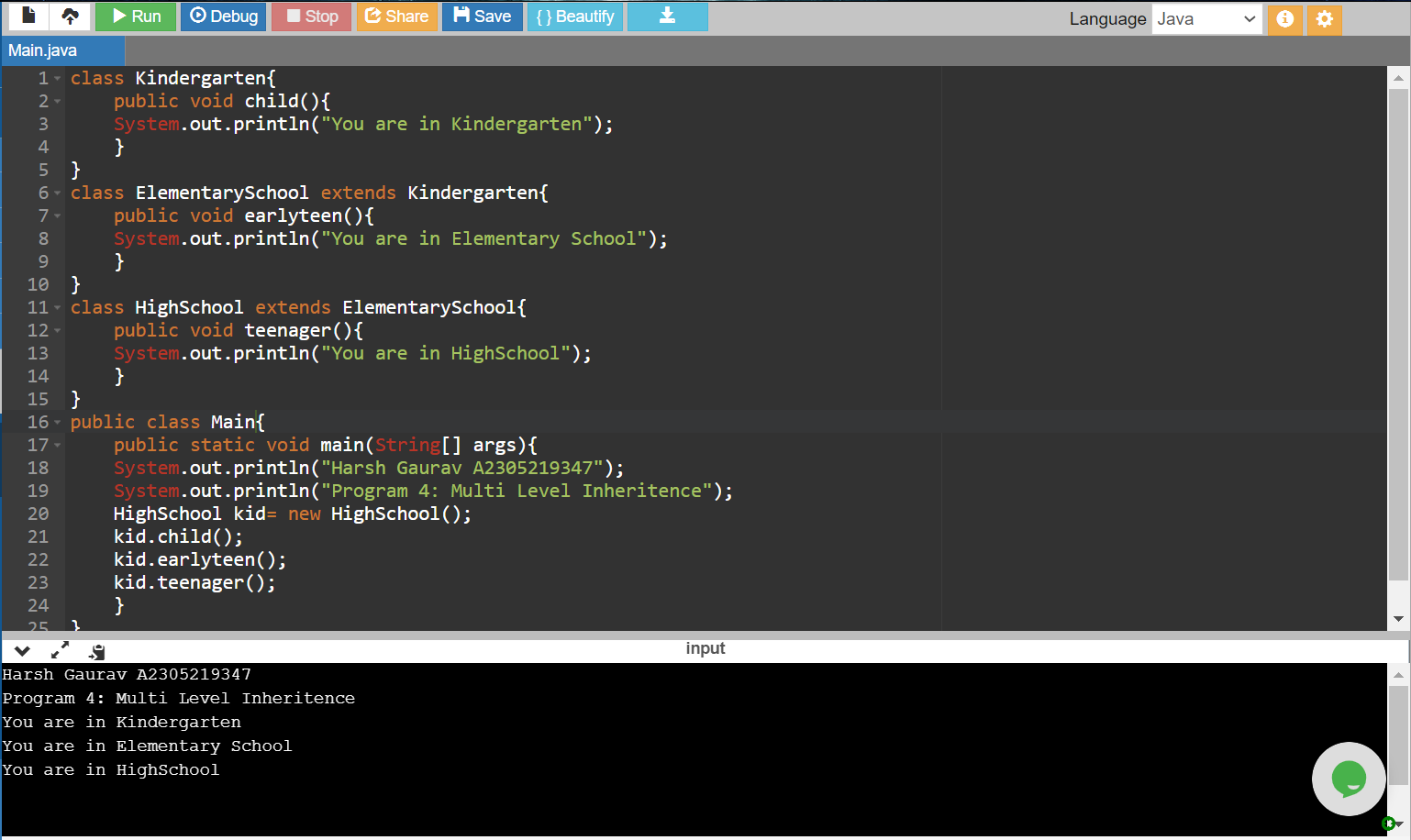
g.display();

s.display();

}

}

Q16. Write a program to show multi-level inheritance.



class Kindergarten{

public void child(){

System.out.println("You are in Kindergarten");

}

}

class ElementarySchool extends Kindergarten{

public void earlyteen(){

System.out.println("You are in Elementary School");

}

}

class HighSchool extends ElementarySchool{

public void teenager(){

System.out.println("You are in HighSchool");

}

}

public class Main{

public static void main(String[] args){

System.out.println("Harsh Gaurav A2305219347");

System.out.println("Program 16: Multi Level Inheritence");

HighSchool kid= new HighSchool();

kid.child();

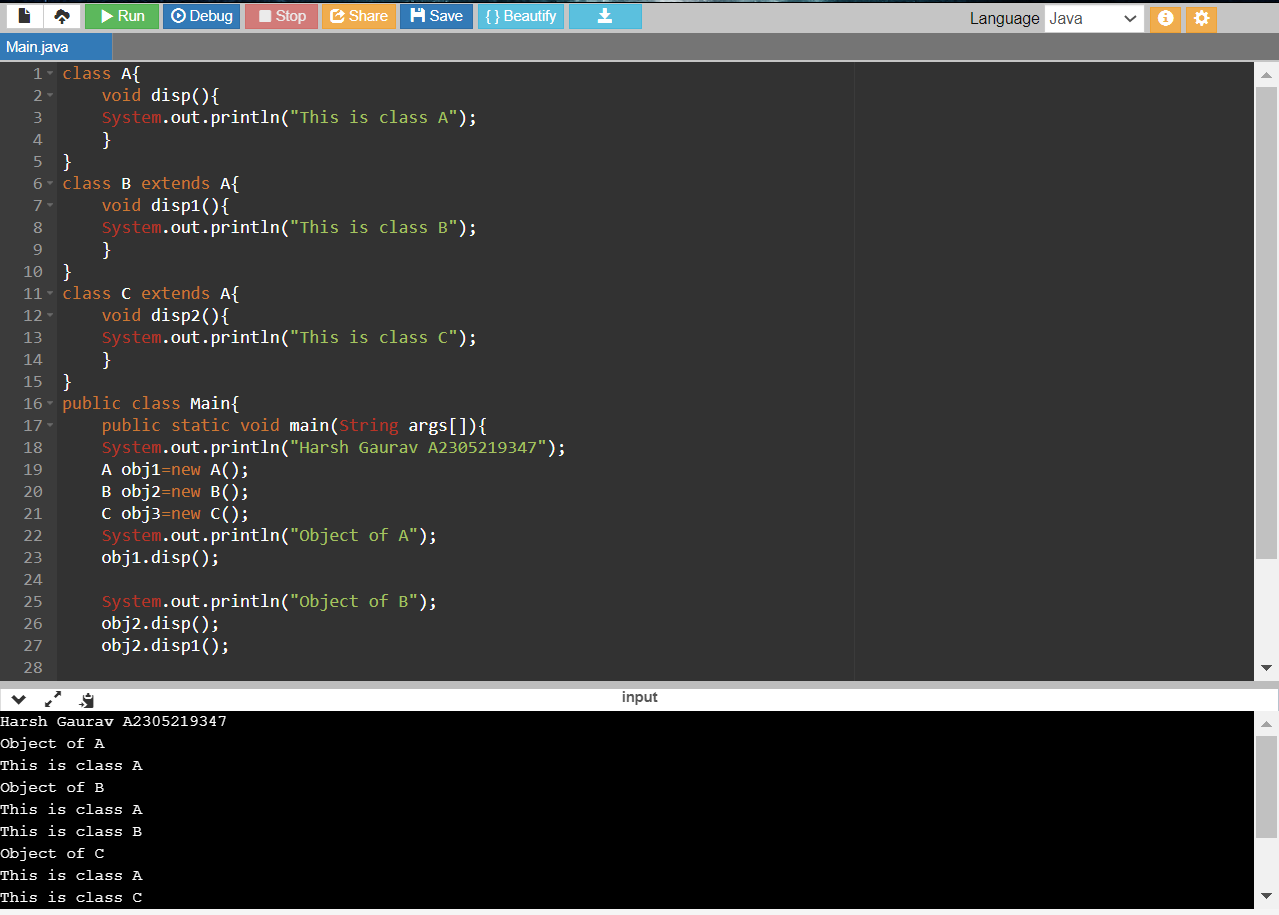
kid.earlyteen();

kid.teenager();

}

}

Q17. Write a program to show hierarchical inheritance.



class A{

void disp(){

System.out.println("This is class A");

}

}

class B extends A{

void disp1(){

System.out.println("This is class B");

}

}

class C extends A{

void disp2(){

System.out.println("This is class C");

}

}

public class Main{

public static void main(String args[]){

System.out.println("Harsh Gaurav A2305219347");

System.out.println("Program 17: Hierarchical Inheritance");

A obj1=new A();

B obj2=new B();

C obj3=new C();

System.out.println("Object of A");

obj1.disp();

System.out.println("Object of B");

obj2.disp();

obj2.disp1();

System.out.println("Object of C");

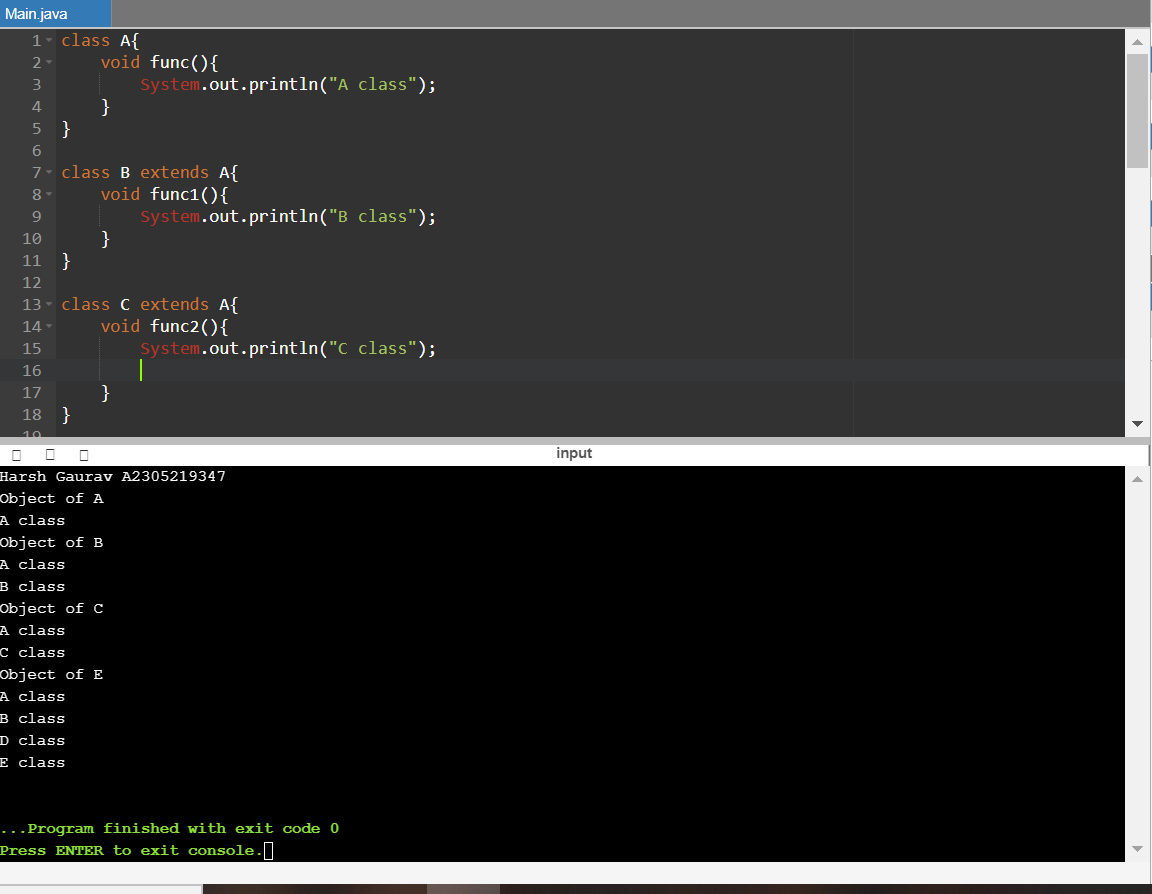
obj3.disp();

obj3.disp2();

}

}

Q18. Write a program to show hybrid inheritance.



class A{

void func(){

System.out.println("A class");

}

}

class B extends A{

void func1(){

System.out.println("B class");

}

}

class C extends A{

void func2(){

System.out.println("C class");

}

}

class D extends B{

void func3(){

System.out.println("D class");

}

}

class E extends D{

void func4(){

System.out.println("E class");

}

}

public class Main

{

public static void main(String[] args) {

System.out.println("Harsh Gaurav A2305219347");

System.out.println("Program 18: Hybrid inheritance");

A obj1=new A();

B obj2=new B();

C obj3=new C();

E obj5=new E();

System.out.println("Object of A");

obj1.func();

System.out.println("Object of B");

obj2.func();

obj2.func1();

System.out.println("Object of C");

obj3.func();

obj3.func2();

System.out.println("Object of E");

obj5.func();

obj5.func1();

obj5.func3();

obj5.func4();

}

}

18/1/2021

Q19. Write a program to show concept of multiple inheritance through implementation of interfaces in a class.



interface Anime

{

void Animation();

}

interface Manga

{

void Drawing();

}

class Japan implements Anime, Manga

{

public void Animation()

{

System.out.println("We make animations to make animes");

}

public void Drawing()

{

System.out.println("We make drawings to publish Manga");

}

}

public class Main

{

public static void main(String[] args) {

System.out.println("Harsh Gaurav A2305219347");

System.out.println("Program 19: Multiple Inheritance thorugh interfaces");

Japan j=new Japan();

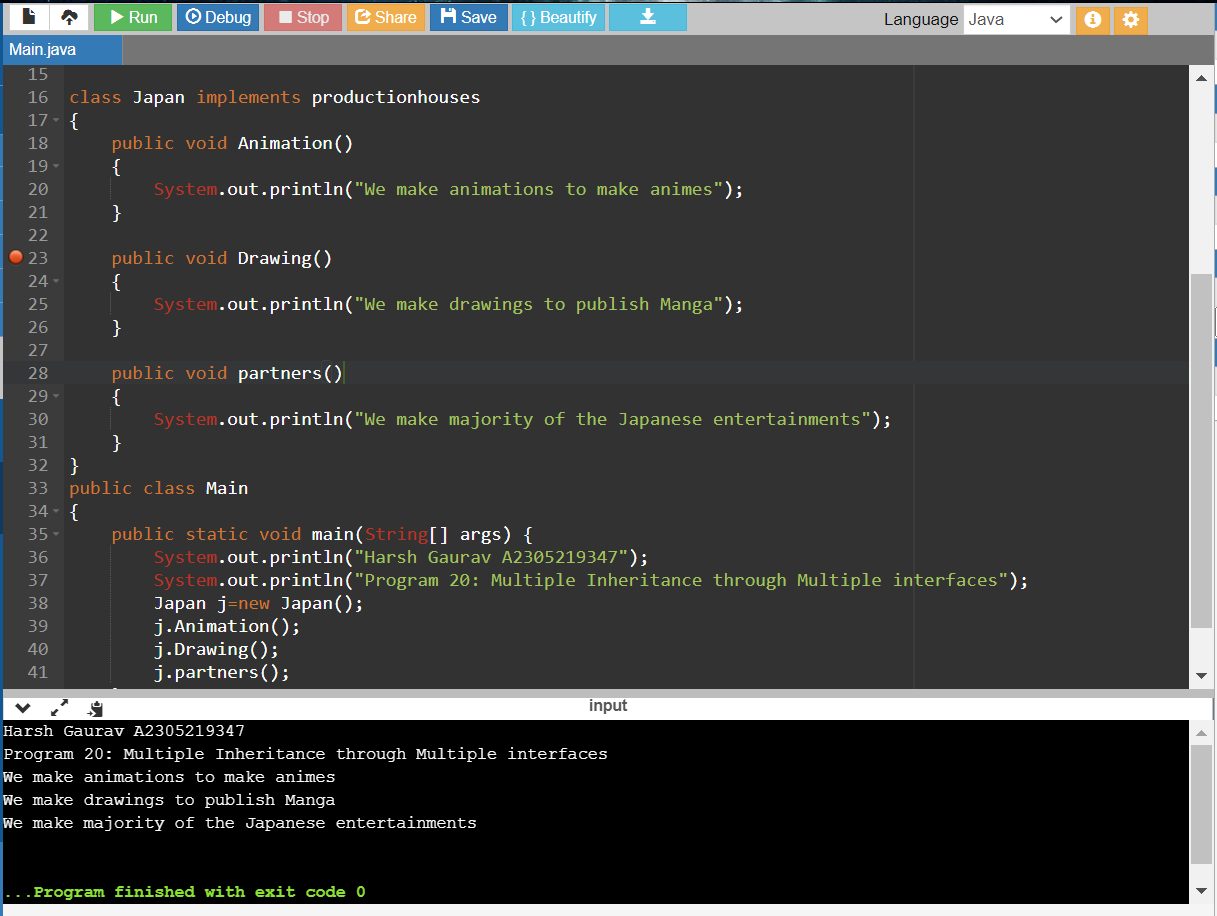
j.Animation();

j.Drawing();

}

}

Q20. Write a program to show concept of multiple inheritance through implementations of interfaces in another that then gets extended in a class.



interface Anime

{

void Animation();

}

interface Manga

{

void Drawing();

}

interface productionhouses extends Anime, Manga

{

public void partners();

}

class Japan implements productionhouses

{

public void Animation()

{

System.out.println("We make animations to make animes");

}

public void Drawing()

{

System.out.println("We make drawings to publish Manga");

}

public void partners()

{

System.out.println("We make majority of the Japanese entertainments");

}

}

public class Main

{

public static void main(String[] args) {

System.out.println("Harsh Gaurav A2305219347");

System.out.println("Program 20: Multiple Inheritance through Multiple interfaces");

Japan j=new Japan();

j.Animation();

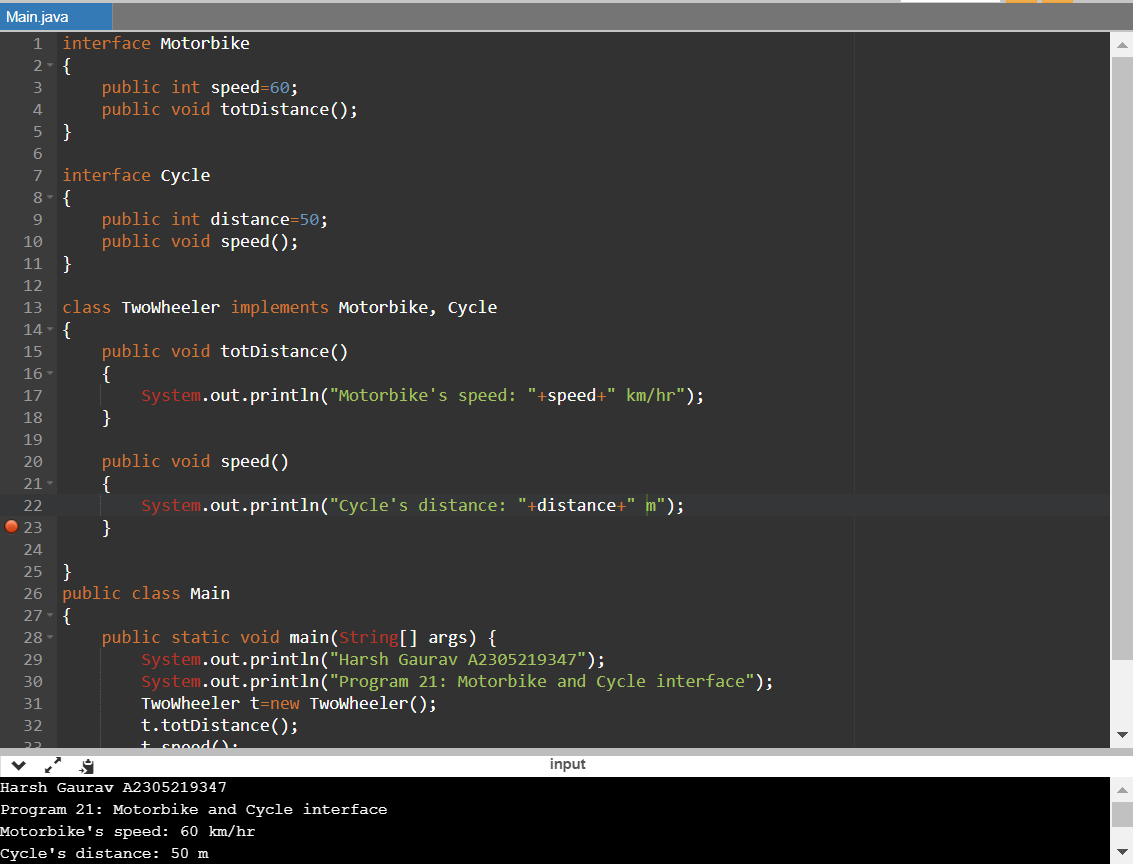
j.Drawing();

j.partners();

}

}

Q21. Write a program with given interfaces Motorbike and Cycle, then implement in child class TwoWheeler and display distance and speed.



interface Motorbike

{

public int speed=60;

public void totDistance();

}

interface Cycle

{

public int distance=50;

public void speed();

}

class TwoWheeler implements Motorbike, Cycle

{

public void totDistance()

{

System.out.println("Motorbike's speed: "+speed+" km/hr");

}

public void speed()

{

System.out.println("Cycle's distance: "+distance+" m");

}

}

public class Main

{

public static void main(String[] args) {

System.out.println("Harsh Gaurav A2305219347");

System.out.println("Program 21: Motorbike and Cycle interface");

TwoWheeler t=new TwoWheeler();

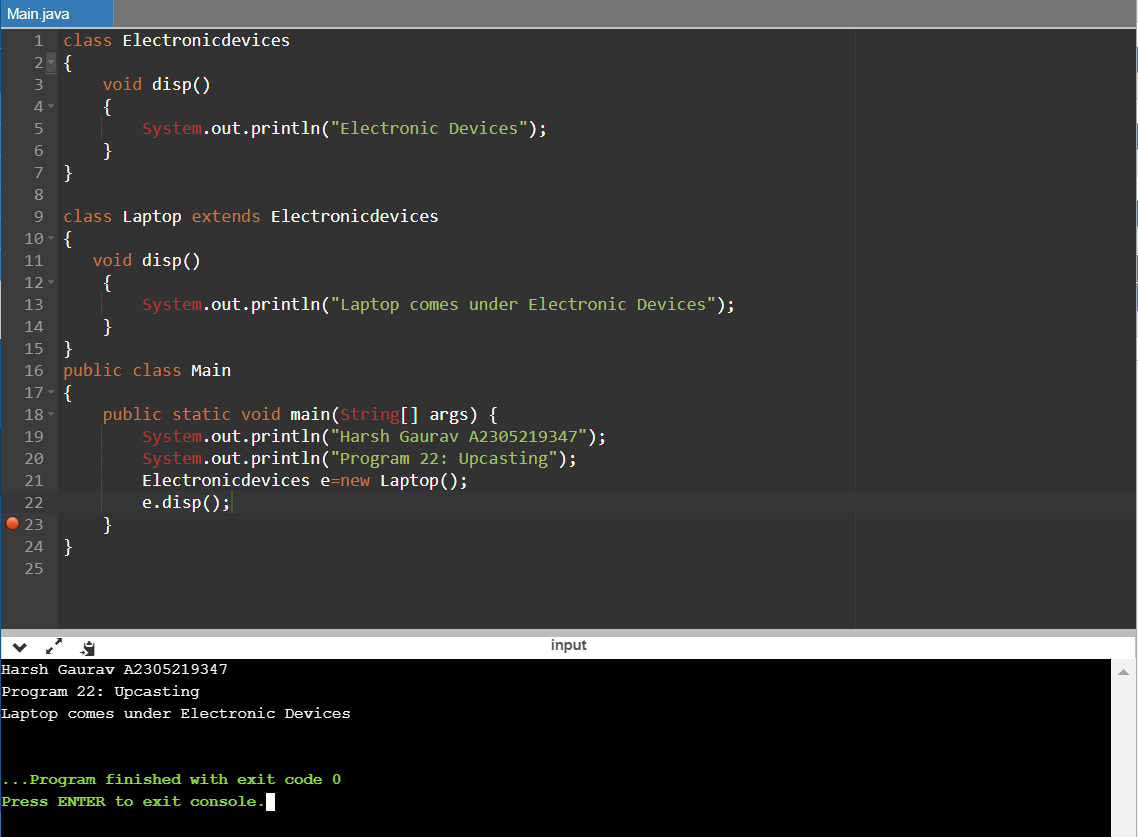
t.totDistance();

t.speed();

}

}

Q22.Write a program to show upcasting in Java.



class Electronicdevices

{

void disp()

{

System.out.println("Electronic Devices");

}

}

class Laptop extends Electronicdevices

{

void disp()

{

System.out.println("Laptop comes under Electronic Devices");

}

}

public class Main

{

public static void main(String[] args) {

System.out.println("Harsh Gaurav A2305219347");

System.out.println("Program 22: Upcasting");

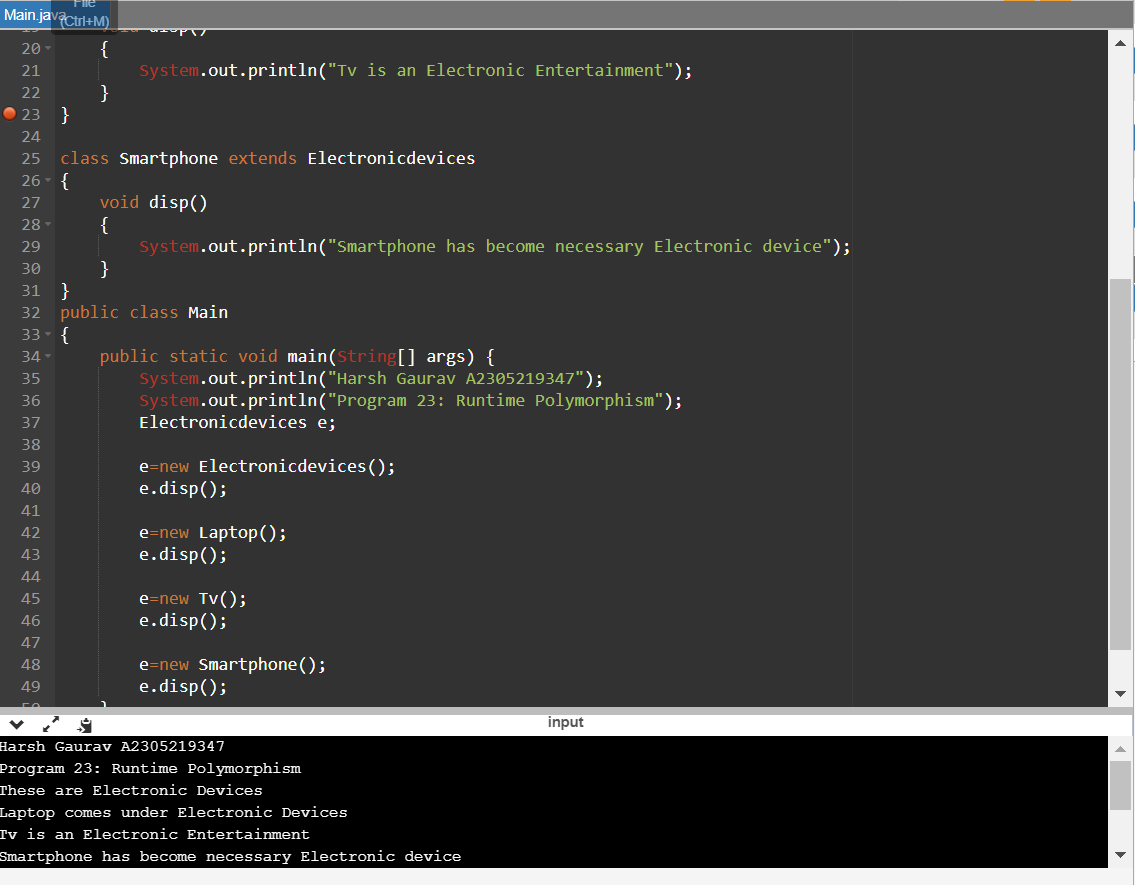
Electronicdevices e=new Laptop();

e.disp();

}

}

Q23. Write a program to implement runtime polymorphism in Java.



class Electronicdevices

{

void disp()

{

System.out.println("These are Electronic Devices");

}

}

class Laptop extends Electronicdevices

{

void disp()

{

System.out.println("Laptop comes under Electronic Devices");

}

}

class Tv extends Electronicdevices

{

void disp()

{

System.out.println("Tv is an Electronic Entertainment");

}

}

class Smartphone extends Electronicdevices

{

void disp()

{

System.out.println("Smartphone has become necessary Electronic device");

}

}

public class Main

{

public static void main(String[] args) {

System.out.println("Harsh Gaurav A2305219347");

System.out.println("Program 23: Runtime Polymorphism");

Electronicdevices e;

e=new Electronicdevices();

e.disp();

e=new Laptop();

e.disp();

e=new Tv();

e.disp();

e=new Smartphone();

e.disp();

}

}

Q24. Write a program to use Lambda Expression



interface Harsh

{

public void myself();

}

public class Main

{

public static void main(String[] args) {

System.out.println("Harsh Gaurav A2305219347");

System.out.println("Program 24: Using Lambda Expression");

int age=19;

Harsh h=()->

{

System.out.println("I am "+age+" years old");

};

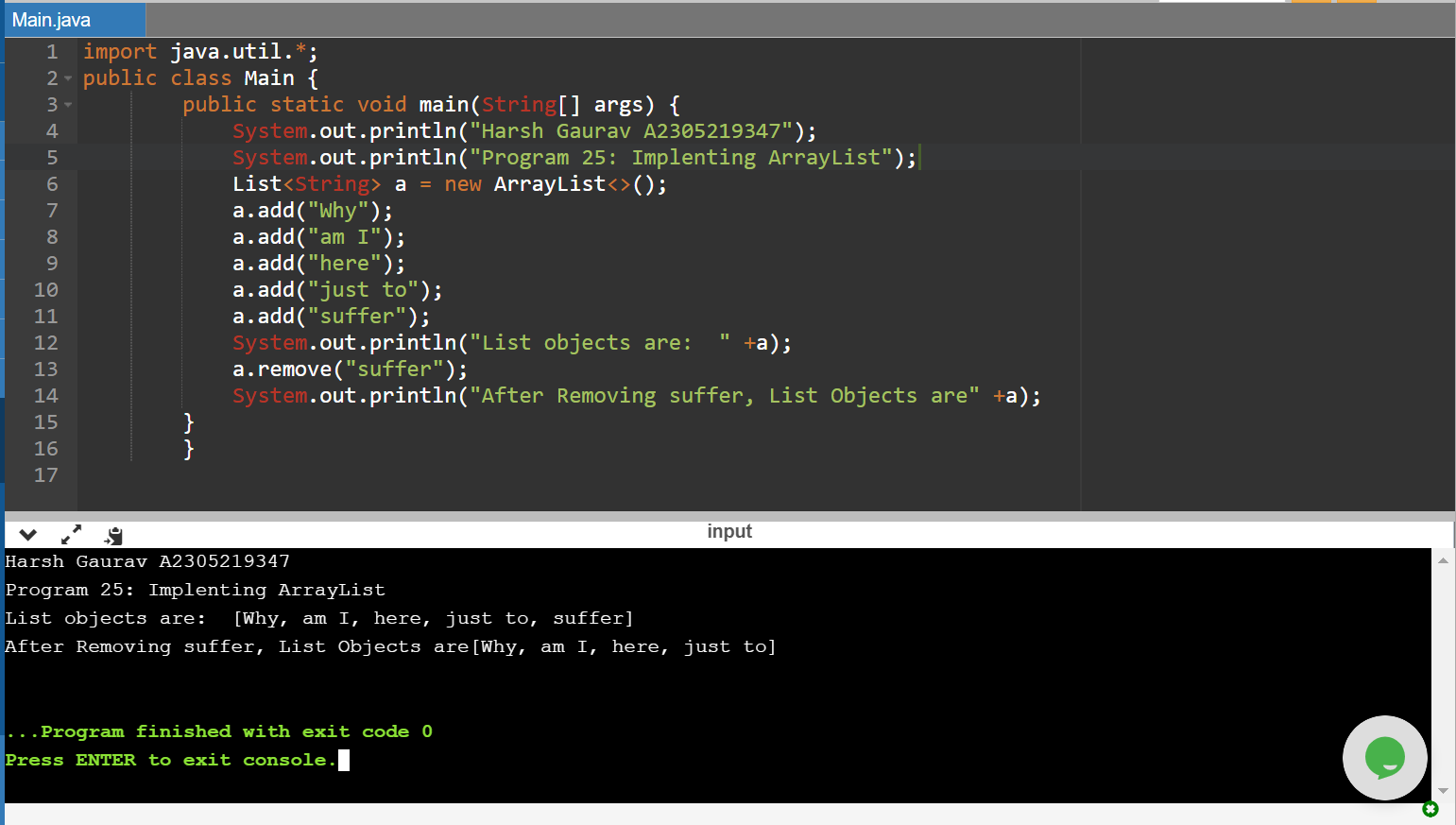
h.myself();

}

}

25/01/2021

Q25. Write a program to implement ArrayList in Java.



import java.util.\*;

public class Main {

public static void main(String[] args) {

System.out.println("Harsh Gaurav A2305219347");

System.out.println("Program 25: Implenting ArrayList");

List<String> a = new ArrayList<>();

a.add("Why");

a.add("am I");

a.add("here");

a.add("just to");

a.add("suffer");

System.out.println("List objects are: " +a);

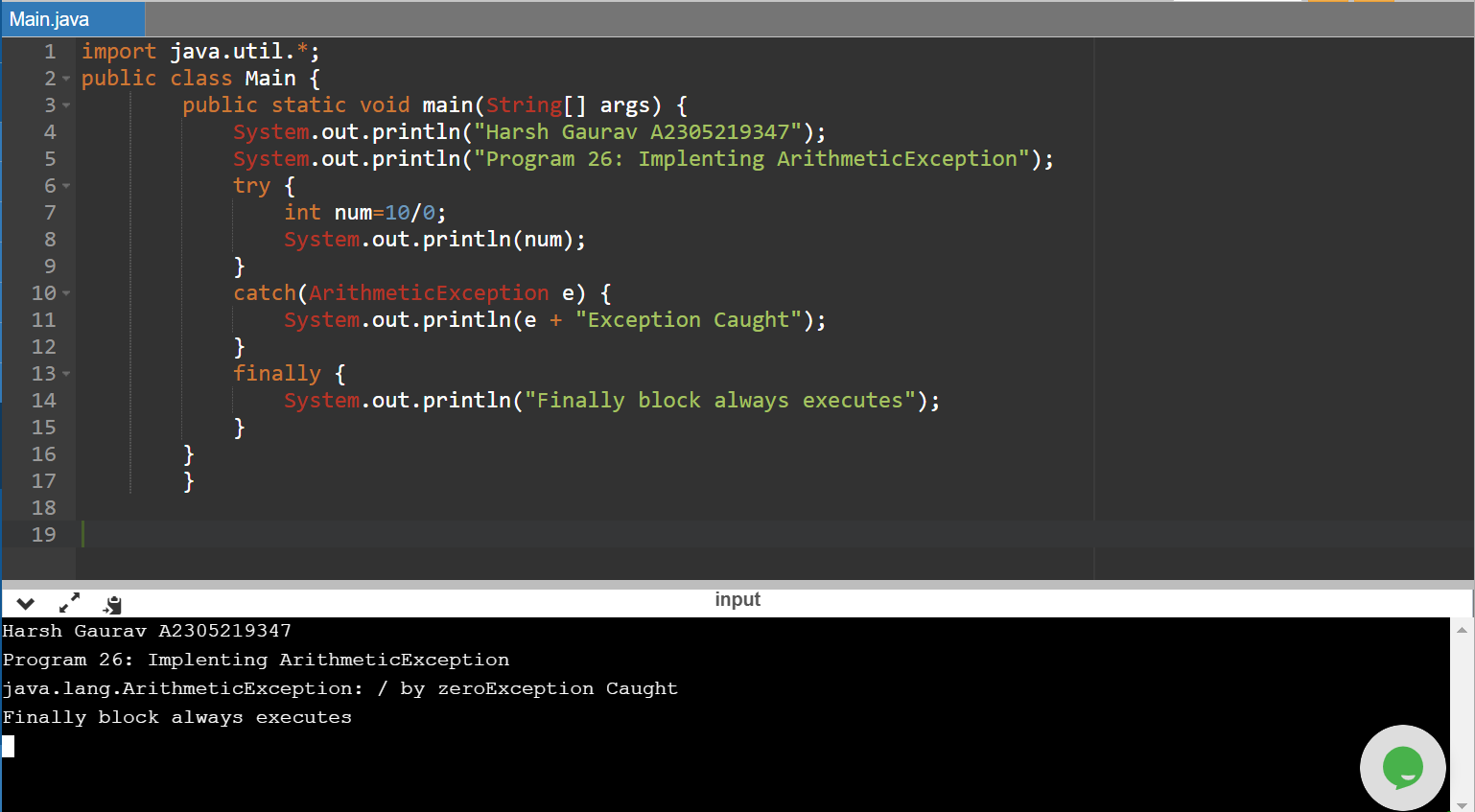
a.remove("suffer");

System.out.println("After Removing suffer, List Objects are" +a);

}

}

Q26. Write a program to show ArithmeticException.



import java.util.\*;

public class Main {

public static void main(String[] args) {

System.out.println("Harsh Gaurav A2305219347");

System.out.println("Program 26: Implenting ArithmeticException");

try {

int num=10/0;

System.out.println(num);

}

catch(ArithmeticException e) {

System.out.println(e + "Exception Caught");

}

finally {

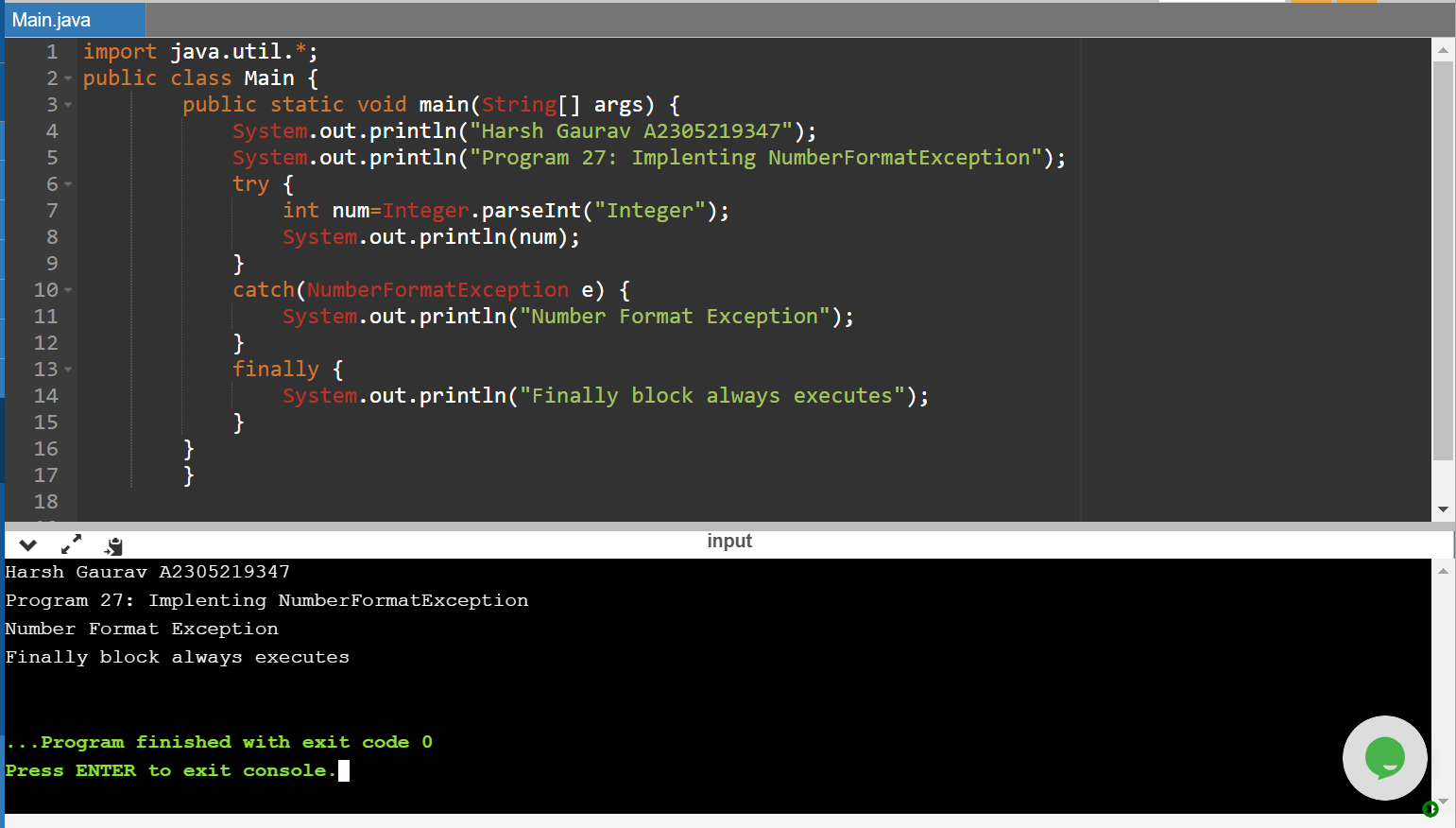
System.out.println("Finally block always executes");

}

}

}

Q27. Write a program to show NumberFormatException.



import java.util.\*;

public class Main {

public static void main(String[] args) {

System.out.println("Harsh Gaurav A2305219347");

System.out.println("Program 27: Implenting NumberFormatException");

try {

int num=Integer.parseInt("Integer");

System.out.println(num);

}

catch(NumberFormatException e) {

System.out.println("Number Format Exception");

}

finally {

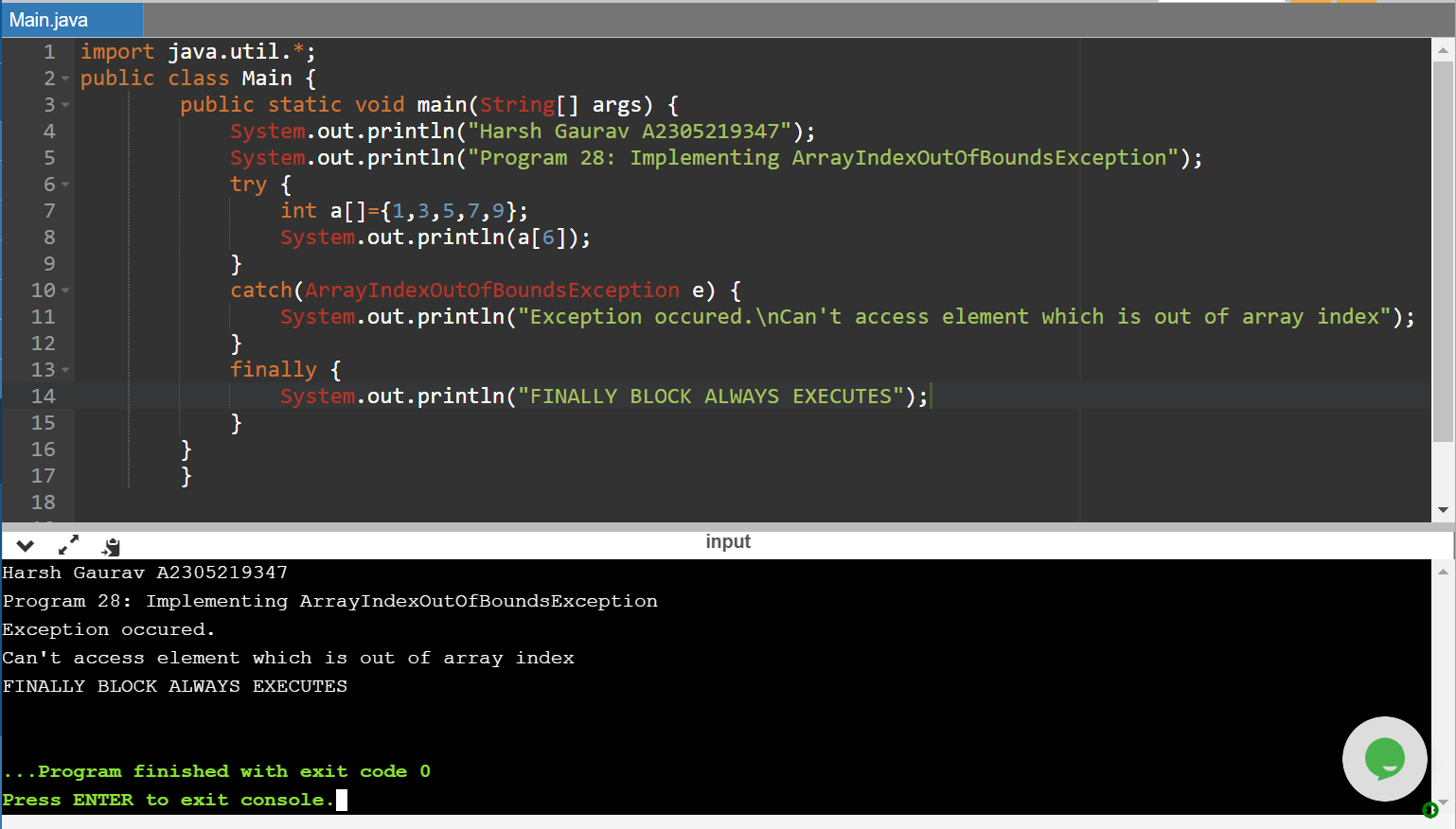
System.out.println("Finally block always executes");

}

}

}

Q28. Write a program to show ArrayIndexOutOfBondsException.



import java.util.\*;

public class Main {

public static void main(String[] args) {

System.out.println("Harsh Gaurav A2305219347");

System.out.println("Program 28: Implementing ArrayIndexOutOfBoundsException");

try {

int a[]={1,3,5,7,9};

System.out.println(a[6]);

}

catch(ArrayIndexOutOfBoundsException e) {

System.out.println("Exception occured.\nCan't access element which is out of array index");

}

finally {

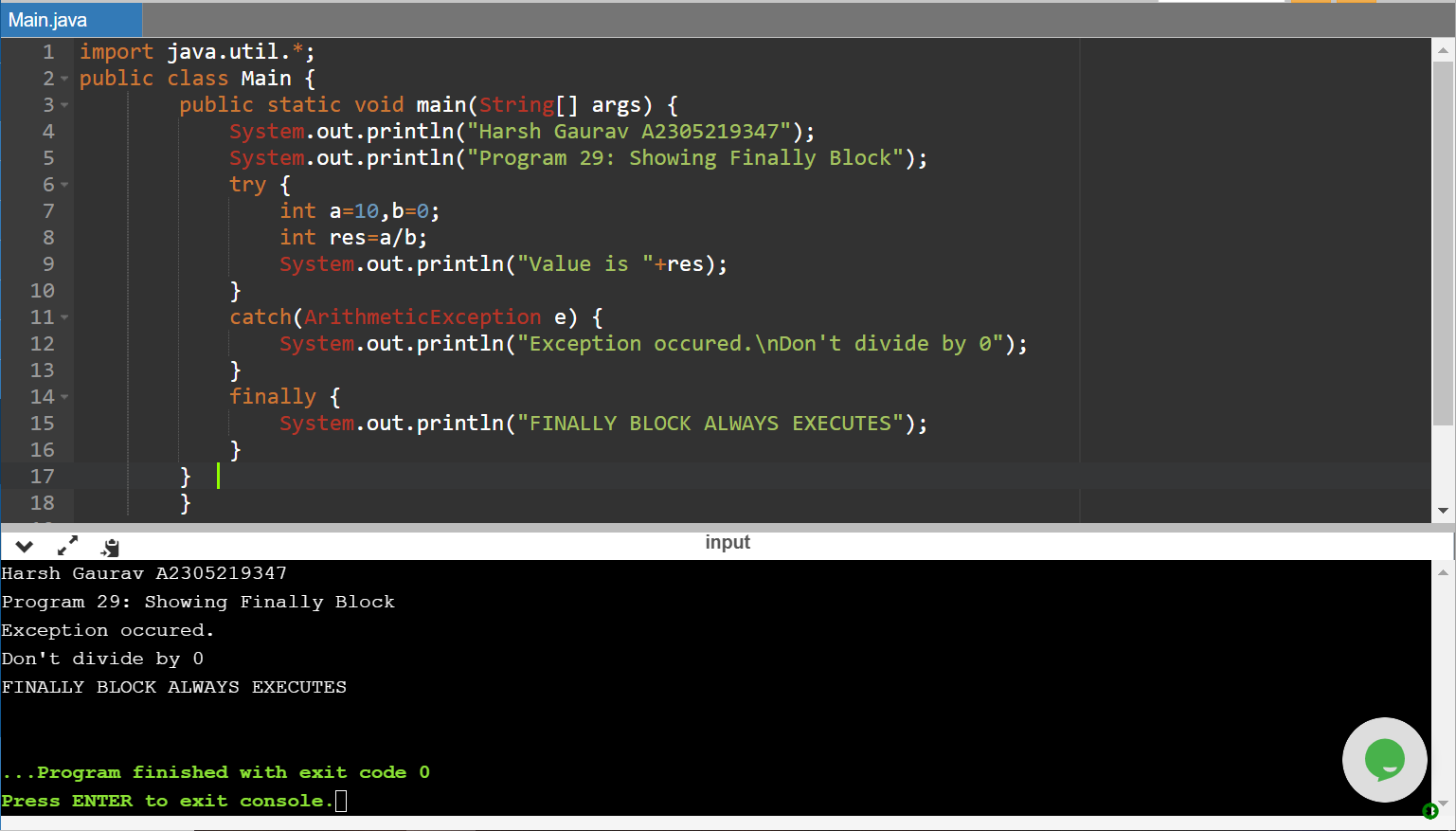
System.out.println("FINALLY BLOCK ALWAYS EXECUTES");

}

}

}

Q29. Write a program to show finally block.



import java.util.\*;

public class Main {

public static void main(String[] args) {

System.out.println("Harsh Gaurav A2305219347");

System.out.println("Program 29: Showing Finally Block");

try {

int a=10,b=0;

int res=a/b;

System.out.println("Value is "+res);

}

catch(ArithmeticException e) {

System.out.println("Exception occured.\nDon't divide by 0");

}

finally {

System.out.println("FINALLY BLOCK ALWAYS EXECUTES");

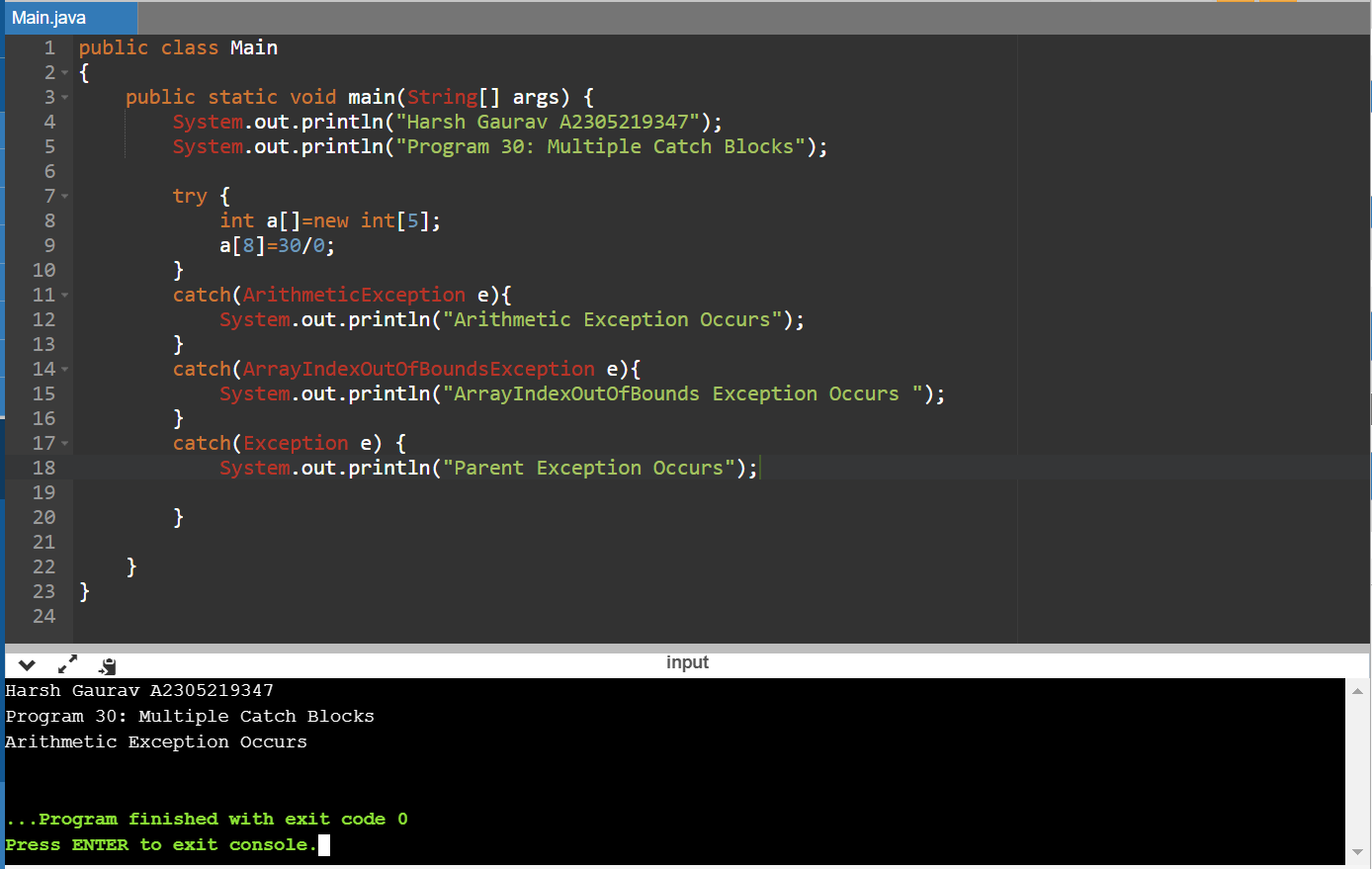
}

}

}

01-02-2021

Q30. Write a program to show multiple catch blocks.



public class Main

{

public static void main(String[] args) {

System.out.println("Harsh Gaurav A2305219347");

System.out.println("Program 30: Multiple Catch Blocks");

try {

int a[]=new int[5];

a[8]=30/0;

}

catch(ArithmeticException e){

System.out.println("Arithmetic Exception Occurs");

}

catch(ArrayIndexOutOfBoundsException e){

System.out.println("ArrayIndexOutOfBounds Exception Occurs ");

}

catch(Exception e) {

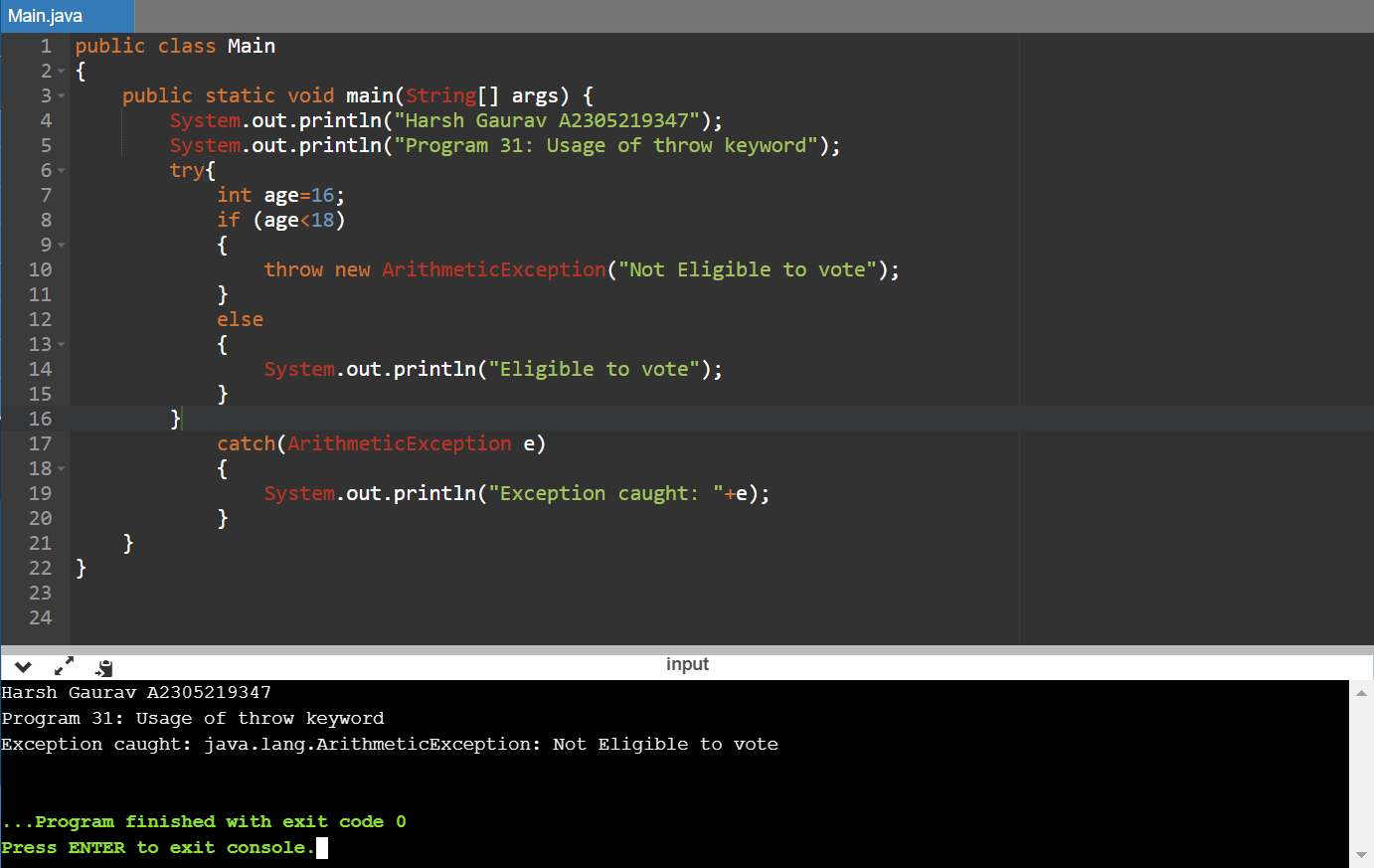
System.out.println("Parent Exception Occurs");

}

}

}

Q31. Write a program to show usage of throw keyword.



public class Main

{

public static void main(String[] args) {

System.out.println("Harsh Gaurav A2305219347");

System.out.println("Program 31: Usage of throw keyword");

try{

int age=16;

if (age<18)

{

throw new ArithmeticException("Not Eligible to vote");

}

else

{

System.out.println("Eligible to vote");

}

}

catch(ArithmeticException e)

{

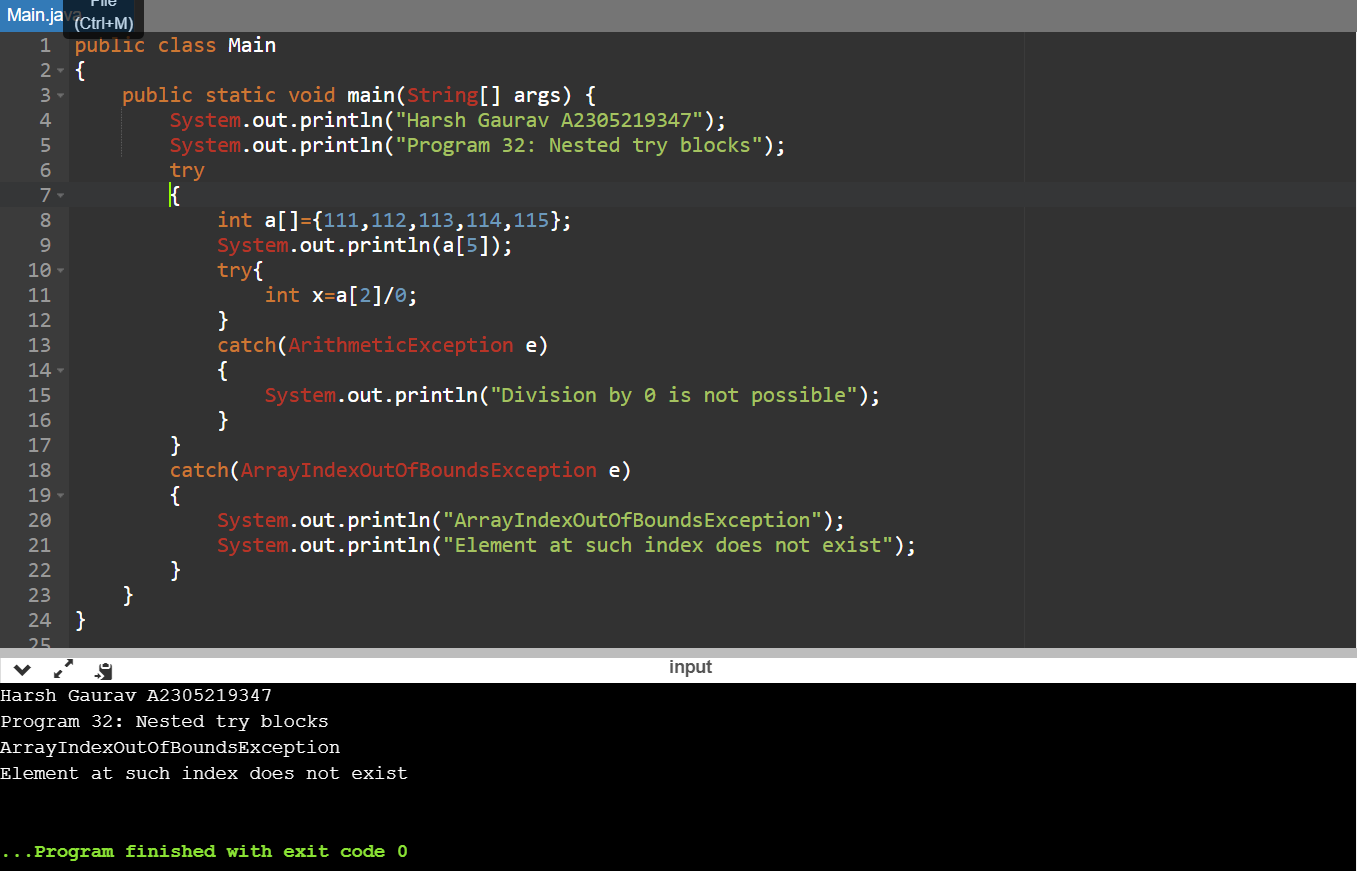
System.out.println("Exception caught: "+e);

}

}

}

Q32. Write a program to show nested try blocks.



public class Main

{

public static void main(String[] args) {

System.out.println("Harsh Gaurav A2305219347");

System.out.println("Program 32: Nested try blocks");

try

{

int a[]={111,112,113,114,115};

System.out.println(a[5]);

try{

int x=a[2]/0;

}

catch(ArithmeticException e)

{

System.out.println("Division by 0 is not possible");

}

}

catch(ArrayIndexOutOfBoundsException e)

{

System.out.println("ArrayIndexOutOfBoundsException");

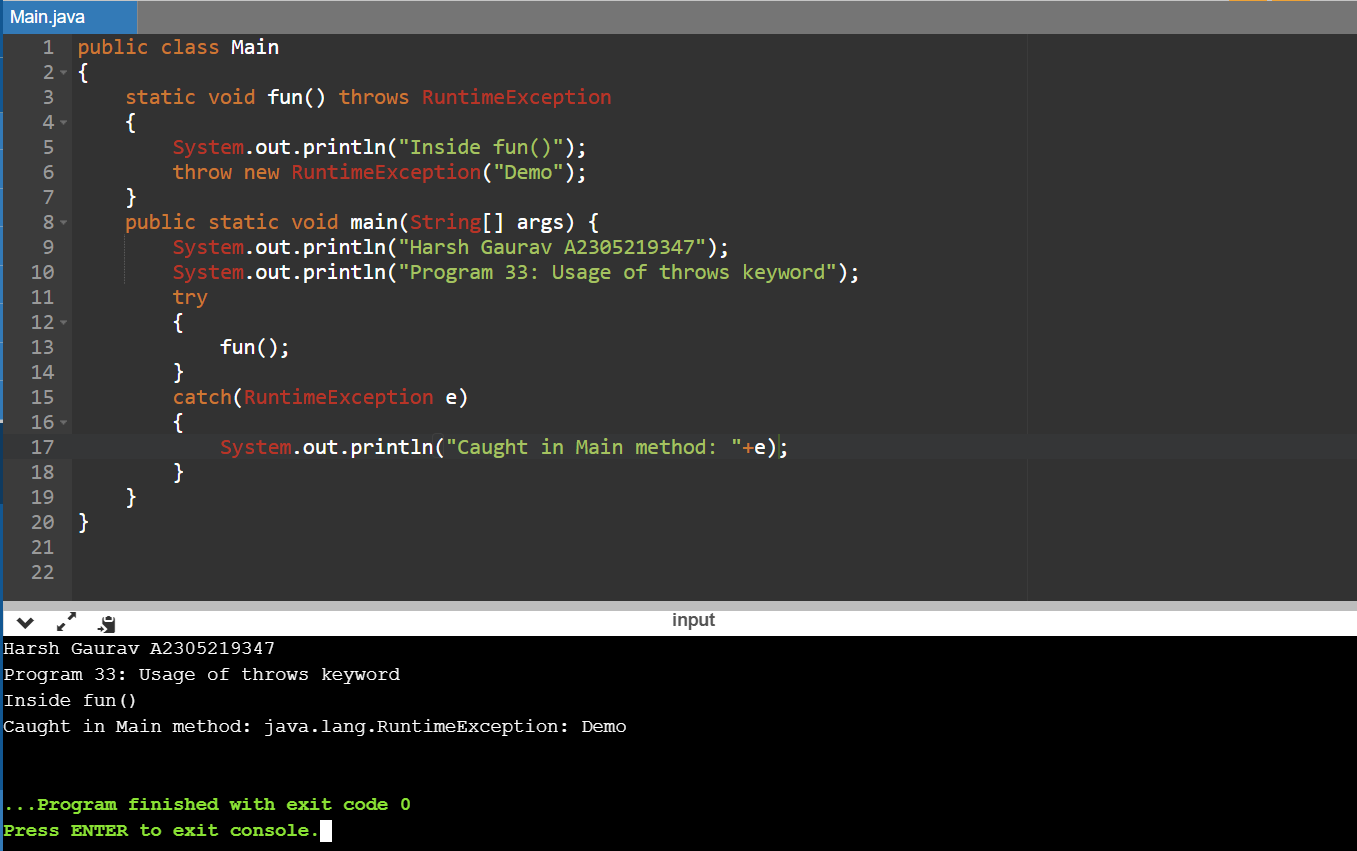
System.out.println("Element at such index does not exist");

}

}

}

Q33. Write a program to show usage of throw keyblocks.



public class Main

{

static void fun() throws RuntimeException

{

System.out.println("Inside fun()");

throw new RuntimeException("Demo");

}

public static void main(String[] args) {

System.out.println("Harsh Gaurav A2305219347");

System.out.println("Program 33: Usage of throws keyword");

try

{

fun();

}

catch(RuntimeException e)

{

System.out.println("Caught in Main method: "+e);

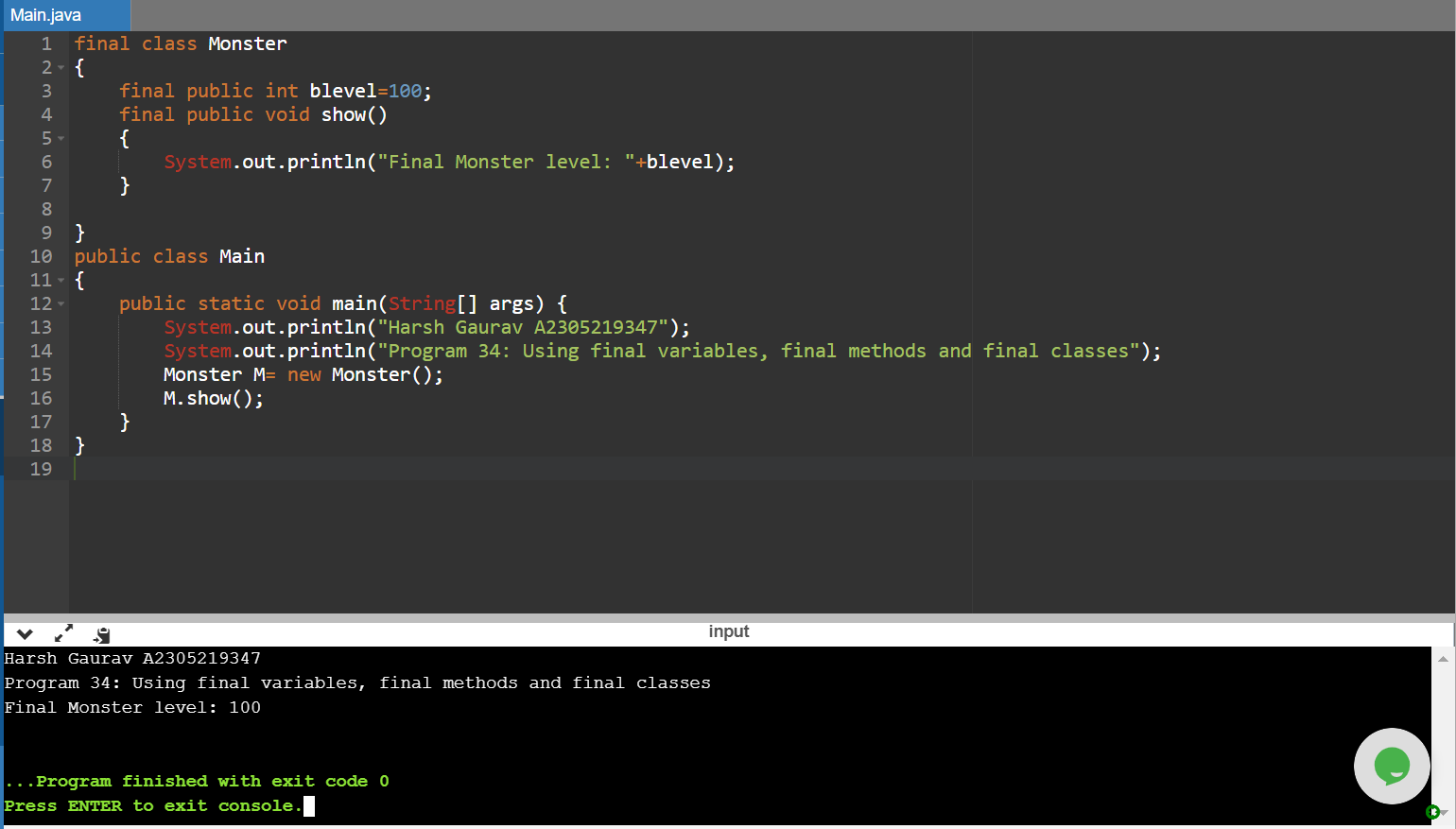
}

}

}

08-02-2021

Q34.Write a program to use final variables, final methods, and final classes.



final class Monster

{

final public int blevel=100;

final public void show()

{

System.out.println("Final Monster level: "+blevel);

}

}

public class Main

{

public static void main(String[] args) {

System.out.println("Harsh Gaurav A2305219347");

System.out.println("Program 34: Using final variables, final methods and final classes");

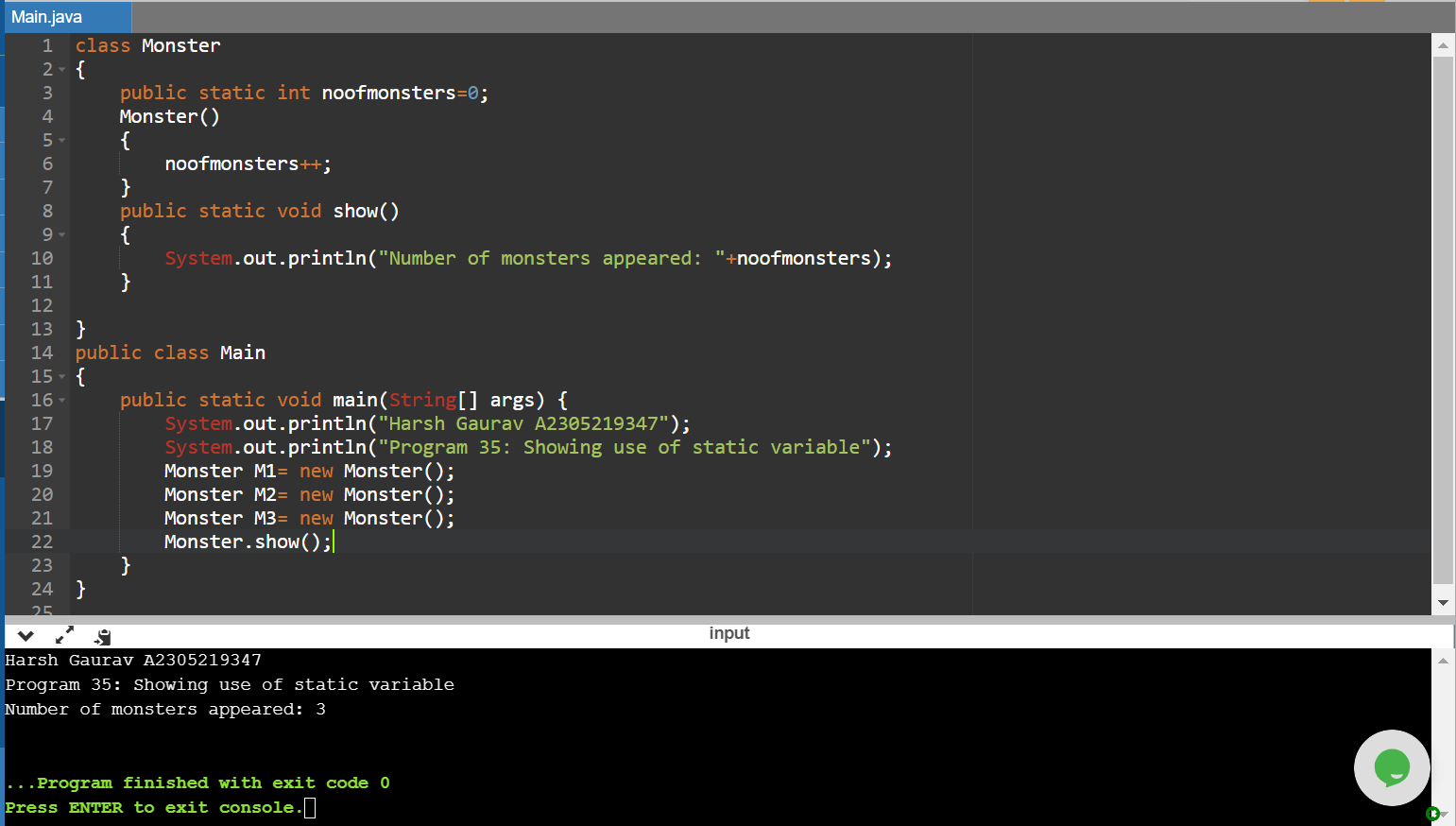
Monster M= new Monster();

M.show();

}

}

Q35. Write a program to show usage of static variable.



class Monster

{

public static int noofmonsters=0;

Monster()

{

noofmonsters++;

}

public static void show()

{

System.out.println("Number of monsters appeared: "+noofmonsters);

}

}

public class Main

{

public static void main(String[] args) {

System.out.println("Harsh Gaurav A2305219347");

System.out.println("Program 35: Showing use of static variable");

Monster M1= new Monster();

Monster M2= new Monster();

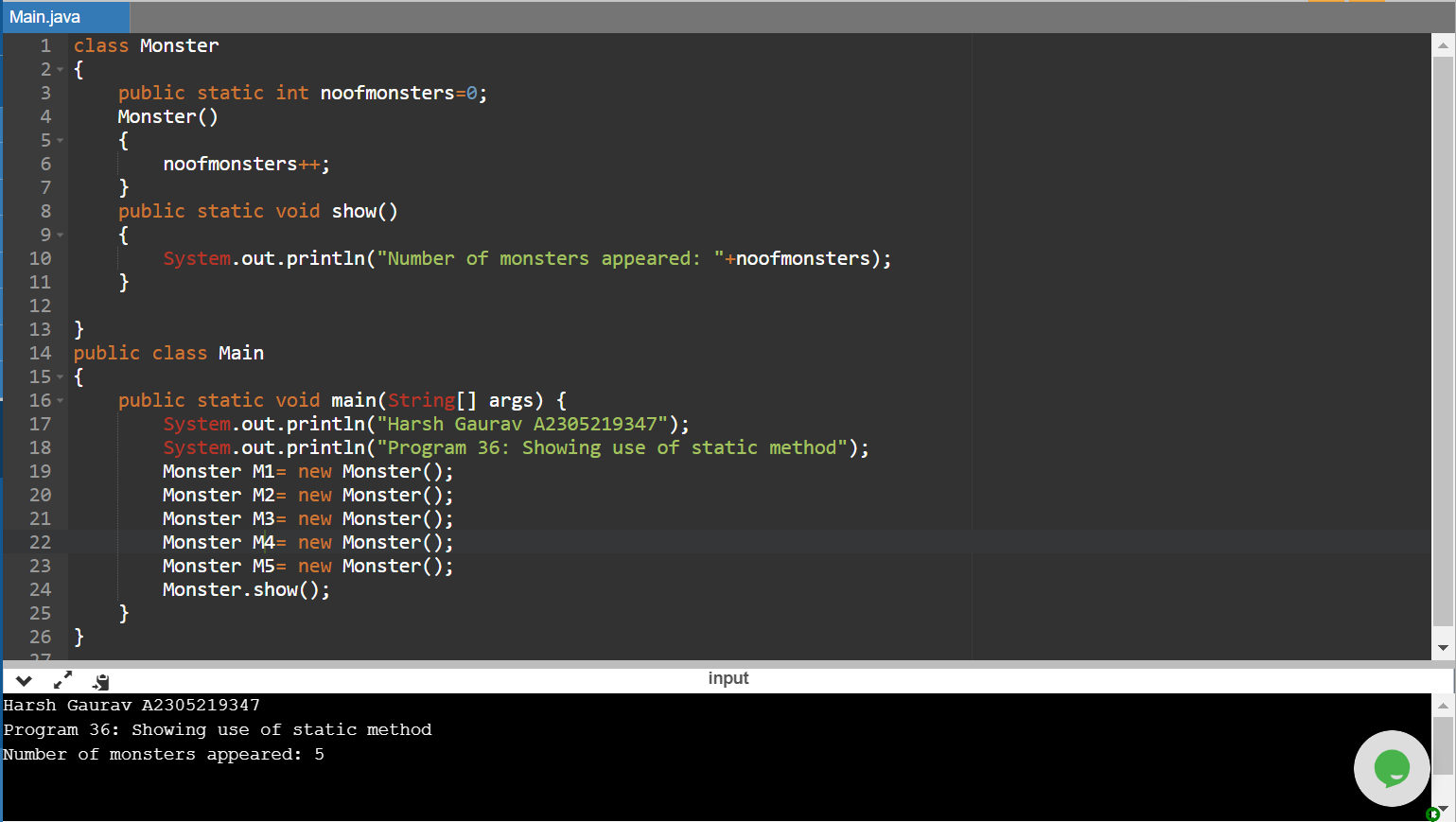
Monster M3= new Monster();

Monster.show();

}

}

Q36. Write a program to show usage of static method.



class Monster

{

public static int noofmonsters=0;

Monster()

{

noofmonsters++;

}

public static void show()

{

System.out.println("Number of monsters appeared: "+noofmonsters);

}

}

public class Main

{

public static void main(String[] args) {

System.out.println("Harsh Gaurav A2305219347");

System.out.println("Program 36: Showing use of static method");

Monster M1= new Monster();

Monster M2= new Monster();

Monster M3= new Monster();

Monster M4= new Monster();

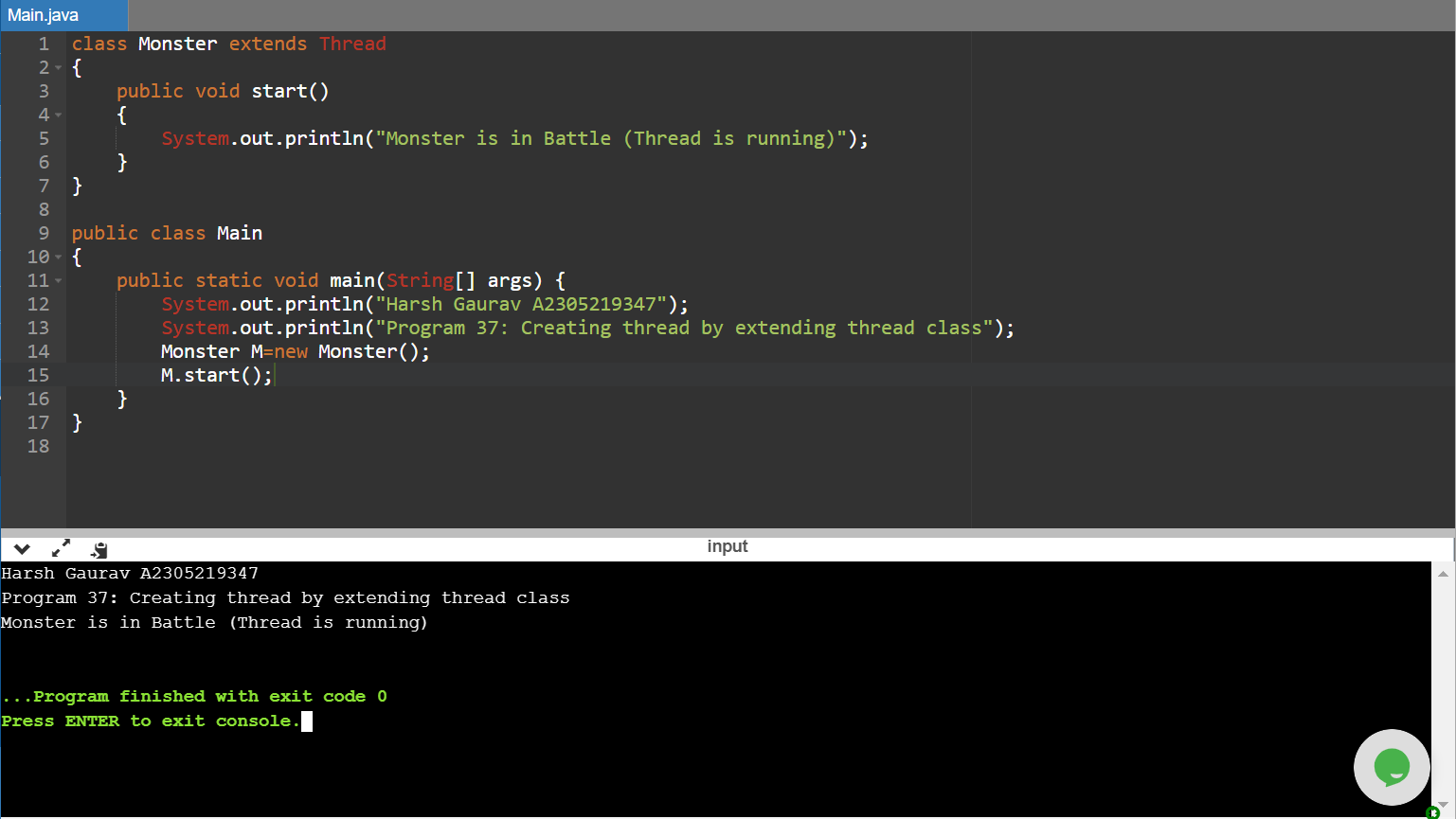
Monster M5= new Monster();

Monster.show();

}

}

Q37. Write a program to create thread by extending Thread class.



class Monster extends Thread

{

public void start()

{

System.out.println("Monster is in Battle (Thread is running)");

}

}

public class Main

{

public static void main(String[] args) {

System.out.println("Harsh Gaurav A2305219347");

System.out.println("Program 37: Creating thread by extending thread class");

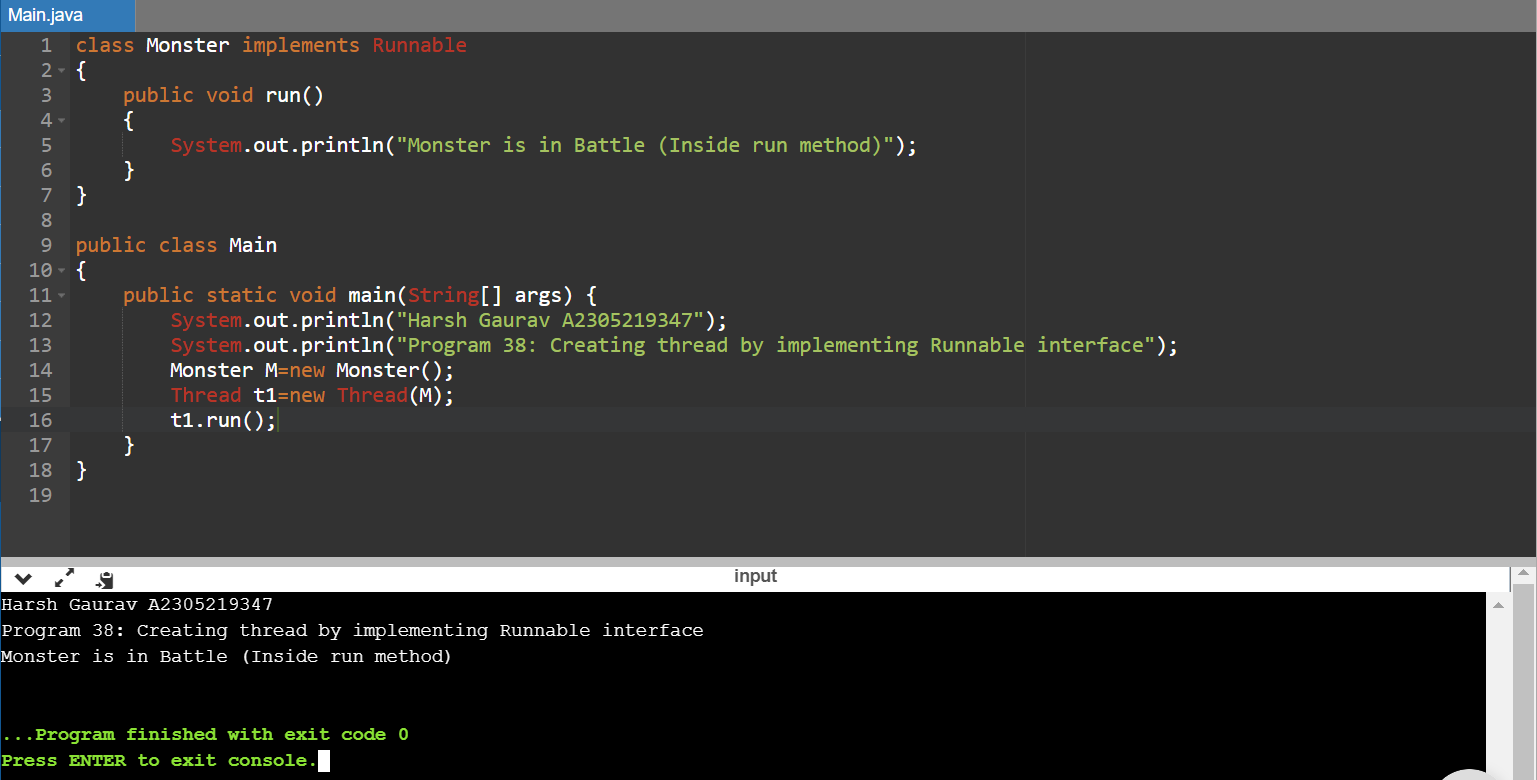
Monster M=new Monster();

M.start();

}

}

Q38. Write a program to create thread by implementing runnable interface.



class Monster implements Runnable

{

public void run()

{

System.out.println("Monster is in Battle (Inside run method)");

}

}

public class Main

{

public static void main(String[] args) {

System.out.println("Harsh Gaurav A2305219347");

System.out.println("Program 38: Creating thread by implementing Runnable interface");

Monster M=new Monster();

Thread t1=new Thread(M);

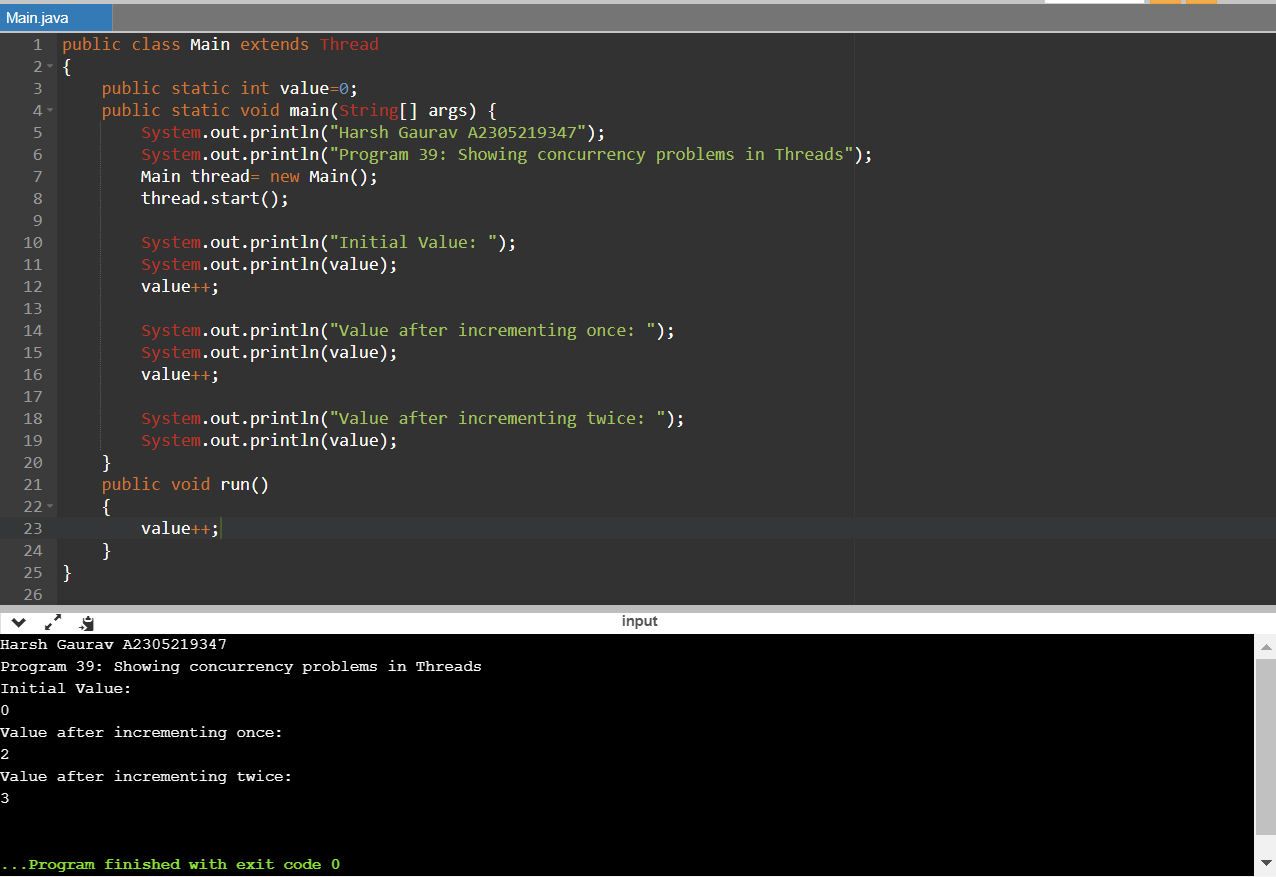
t1.run();

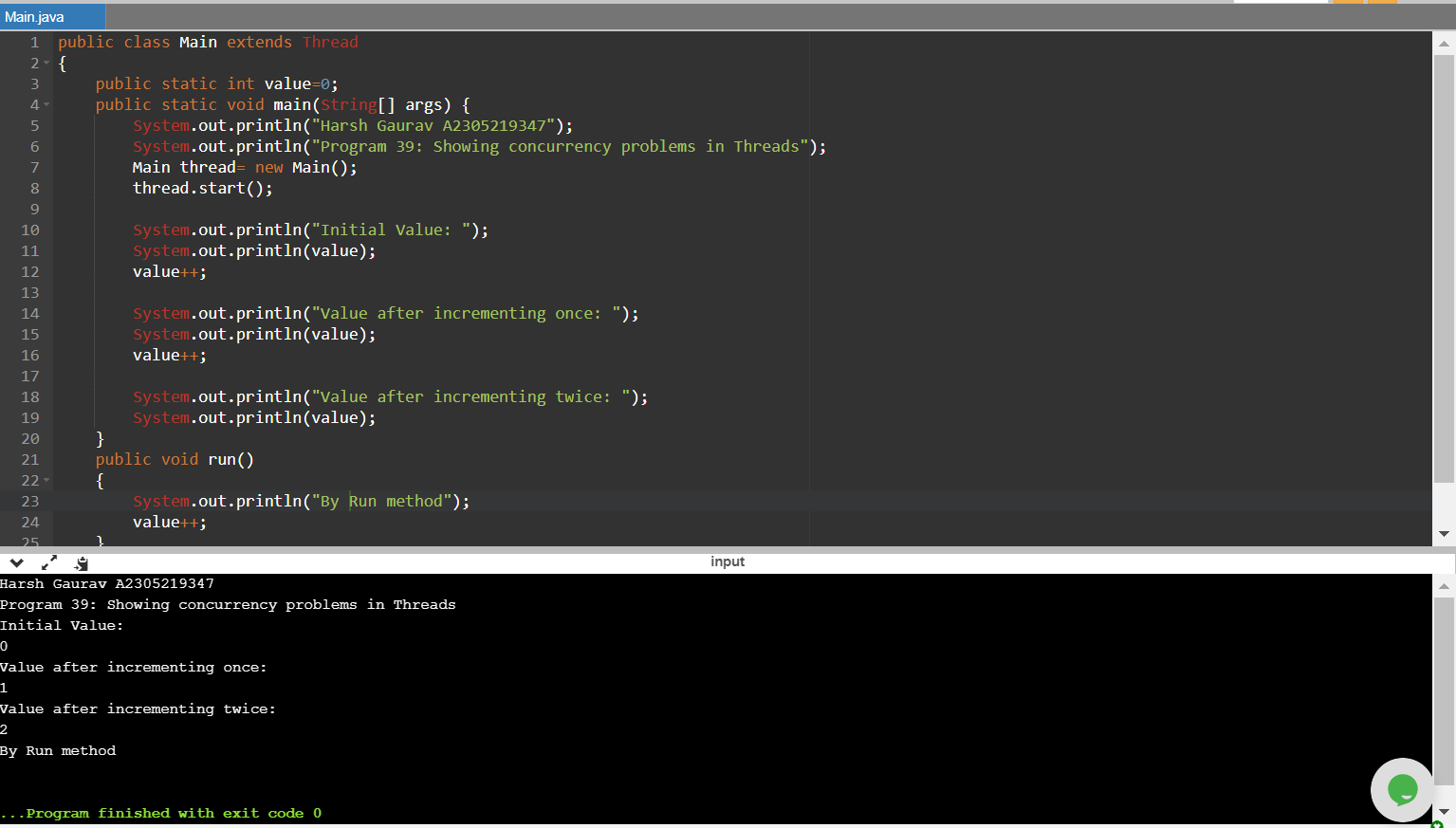
}

}

15-02-2021

Q39. Write a program to show concurrency problems in threads.





public class Main extends Thread

{

public static int value=0;

public static void main(String[] args) {

System.out.println("Harsh Gaurav A2305219347");

System.out.println("Program 39: Showing concurrency problems in Threads");

Main thread= new Main();

thread.start();

System.out.println("Initial Value: ");

System.out.println(value);

value++;

System.out.println("Value after incrementing once: ");

System.out.println(value);

value++;

System.out.println("Value after incrementing twice: ");

System.out.println(value);

}

public void run()

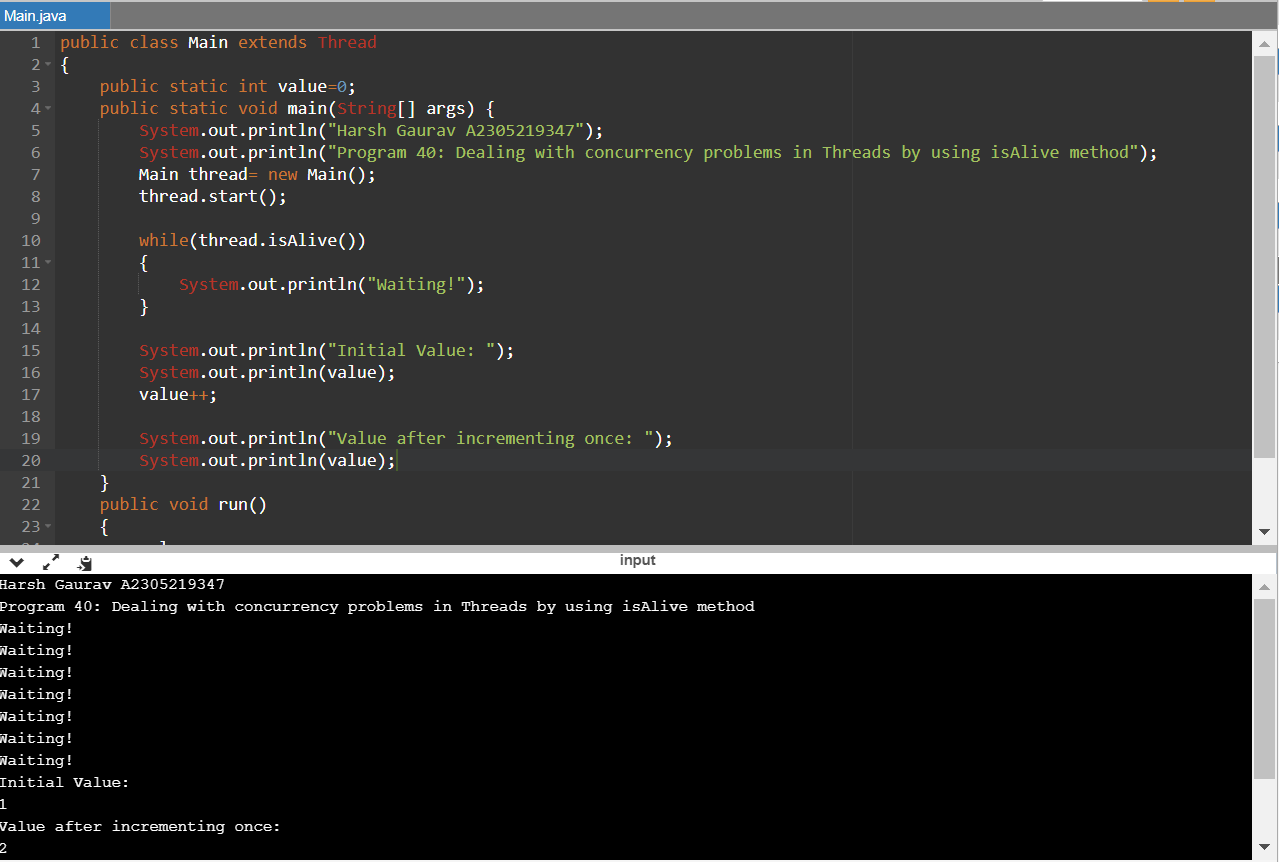
{

value++;

}

}

Q40. Write a program to deal with concurrency problem in threads using isAlive() method.



public class Main extends Thread

{

public static int value=0;

public static void main(String[] args) {

System.out.println("Harsh Gaurav A2305219347");

System.out.println("Program 40: Dealing with concurrency problems in Threads by using isAlive method");

Main thread= new Main();

thread.start();

while(thread.isAlive())

{

System.out.println("Waiting!");

}

System.out.println("Initial Value: ");

System.out.println(value);

value++;

System.out.println("Value after incrementing once: ");

System.out.println(value);

}

public void run()

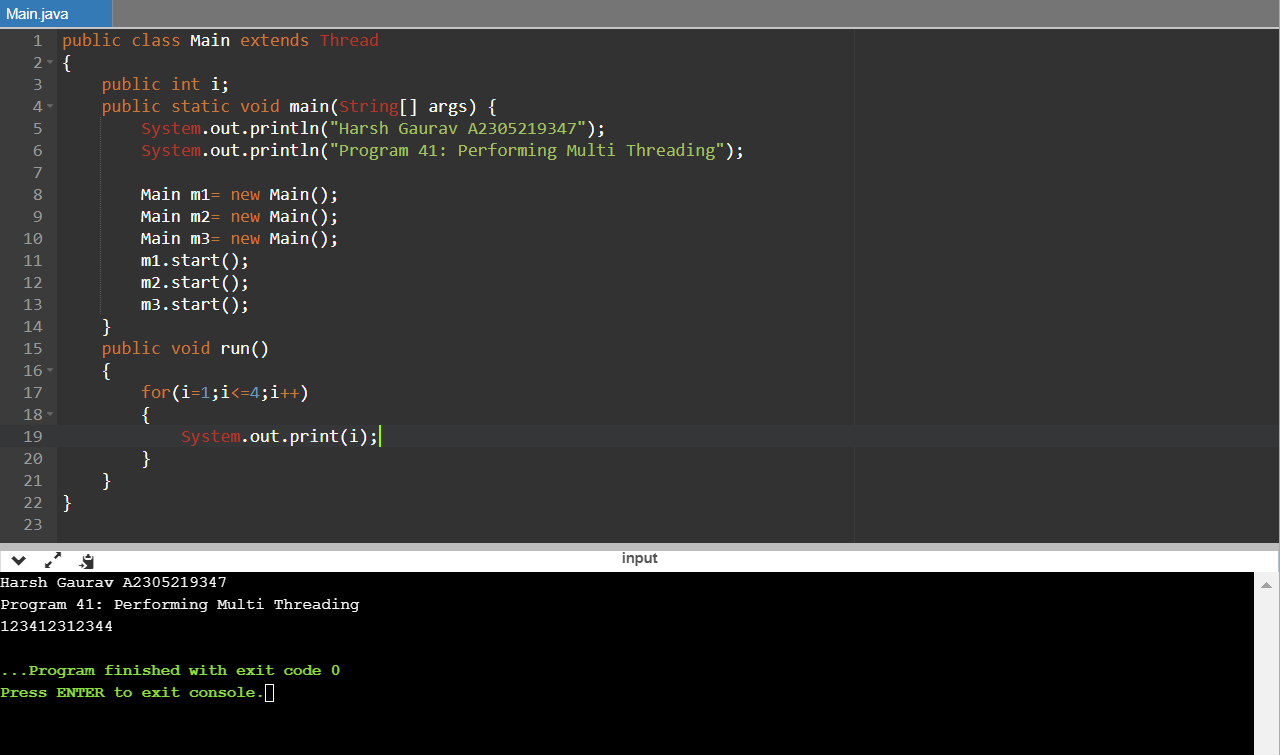
{

value++;

}

}

Q41. Write a program to perform multi-threading.



public class Main extends Thread

{

public int i;

public static void main(String[] args) {

System.out.println("Harsh Gaurav A2305219347");

System.out.println("Program 41: Performing Multi Threading");

Main m1= new Main();

Main m2= new Main();

Main m3= new Main();

m1.start();

m2.start();

m3.start();

}

public void run()

{

for(i=1;i<=4;i++)

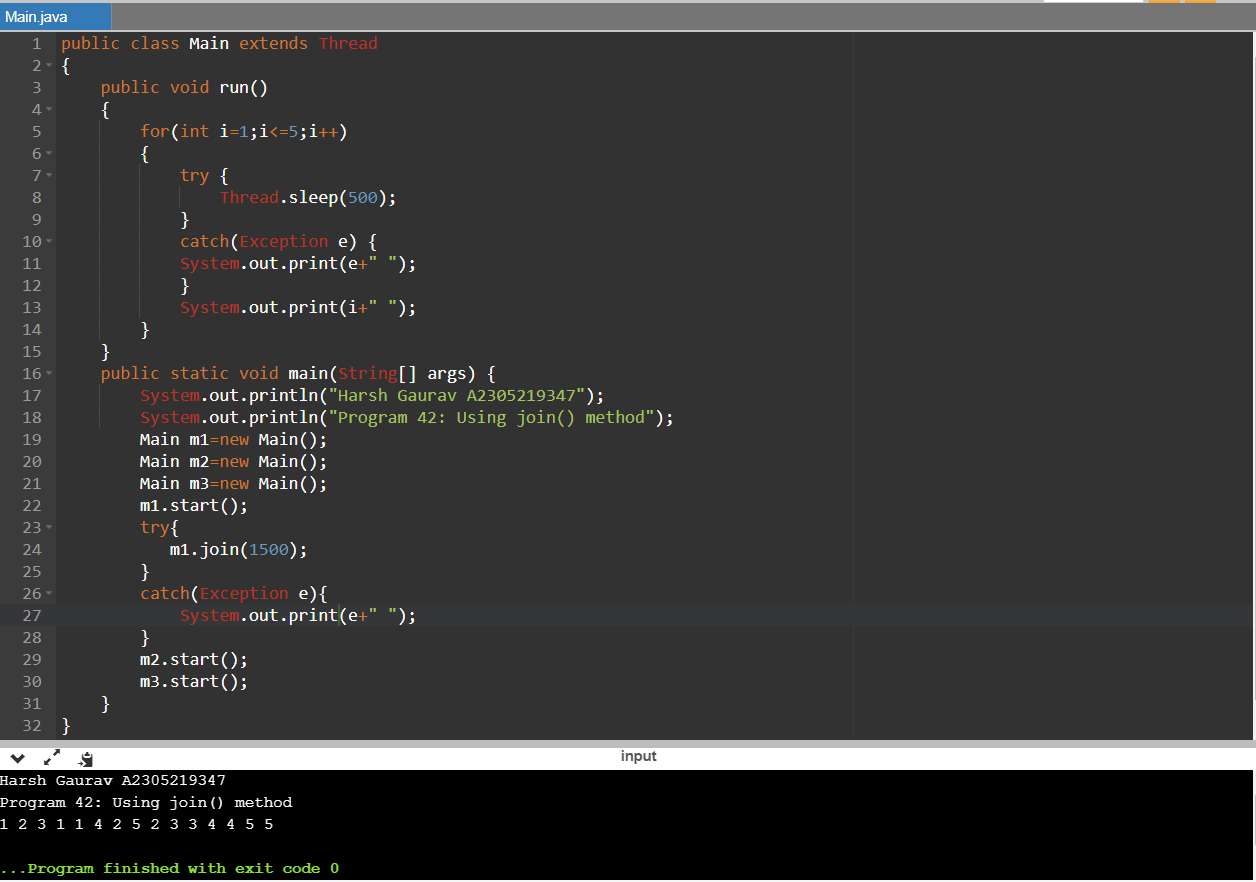
{

System.out.print(i);

}

}

}

Q42. Write a program to show use of join() method.

public class Main extends Thread

{

public void run()

{

for(int i=1;i<=5;i++)

{

try {

Thread.sleep(500);

}

catch(Exception e) {

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

}

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

}

}

public static void main(String[] args) {

System.out.println("Harsh Gaurav A2305219347");

System.out.println("Program 42: Using join() method");

Main m1=new Main();

Main m2=new Main();

Main m3=new Main();

m1.start();

try{

m1.join(1500);

}

catch(Exception e){

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

}

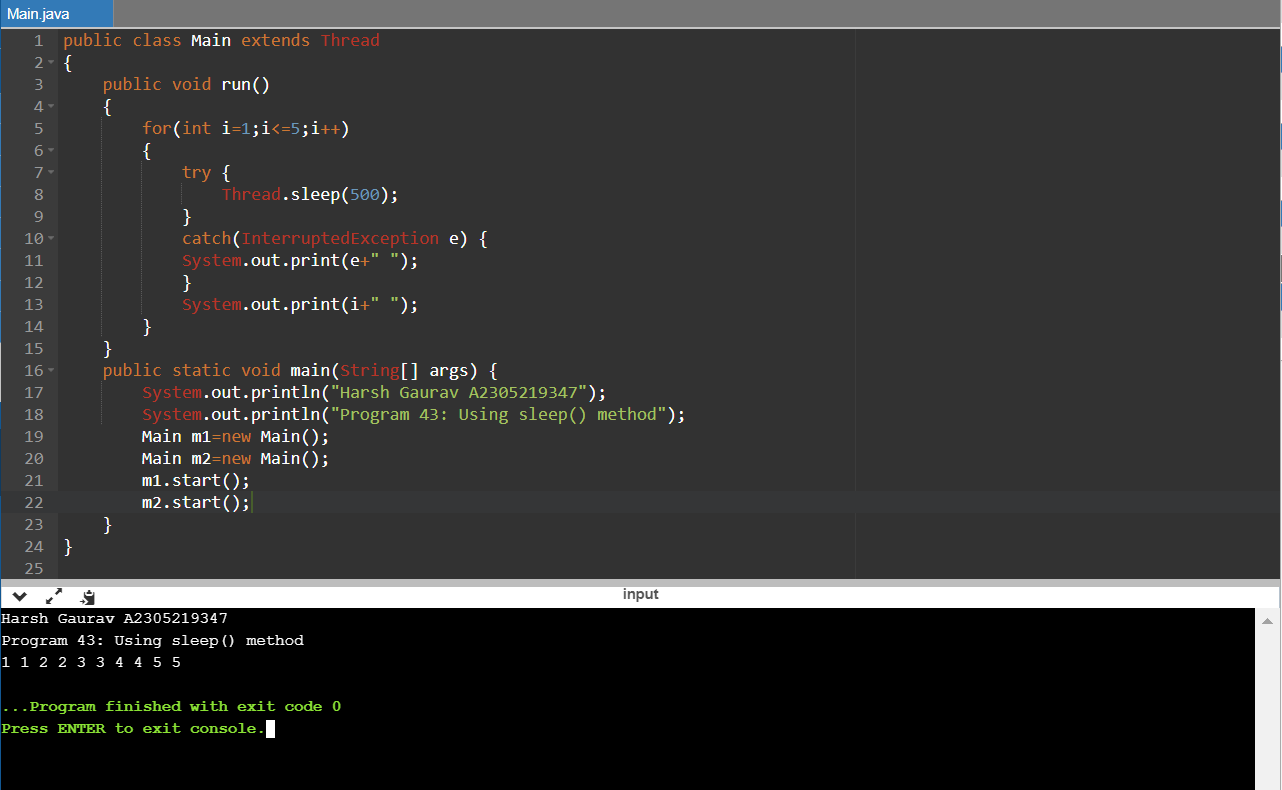
m2.start();

m3.start();

}

}

Q43. Write a program in Java to show use of sleep() method.



public class Main extends Thread

{

public void run()

{

for(int i=1;i<=5;i++)

{

try {

Thread.sleep(500);

}

catch(InterruptedException e) {

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

}

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

}

}

public static void main(String[] args) {

System.out.println("Harsh Gaurav A2305219347");

System.out.println("Program 43: Using sleep() method");

Main m1=new Main();

Main m2=new Main();

m1.start();

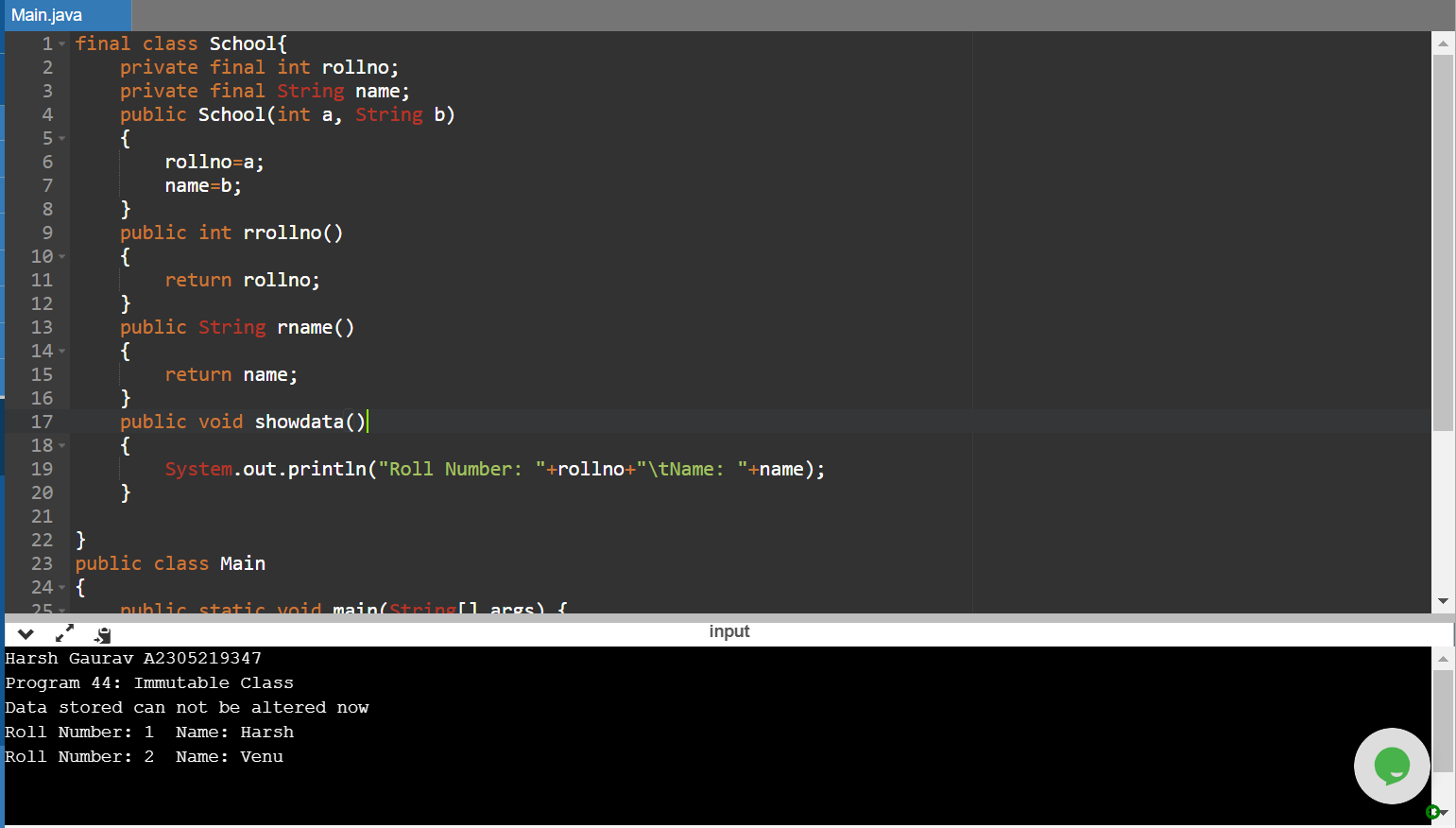
m2.start();

}

}

22-02-2021

Q44. Write a program to create an immutable class.



final class School{

private final int rollno;

private final String name;

public School(int a, String b)

{

rollno=a;

name=b;

}

public int rrollno()

{

return rollno;

}

public String rname()

{

return name;

}

public void showdata()

{

System.out.println("Roll Number: "+rollno+"\tName: "+name);

}

}

public class Main

{

public static void main(String[] args) {

System.out.println("Harsh Gaurav A2305219347");

System.out.println("Program 44: Immutable Class");

System.out.println("Data stored can not be altered now");

School s1=new School(1,"Harsh");

School s2=new School(2,"Venu");

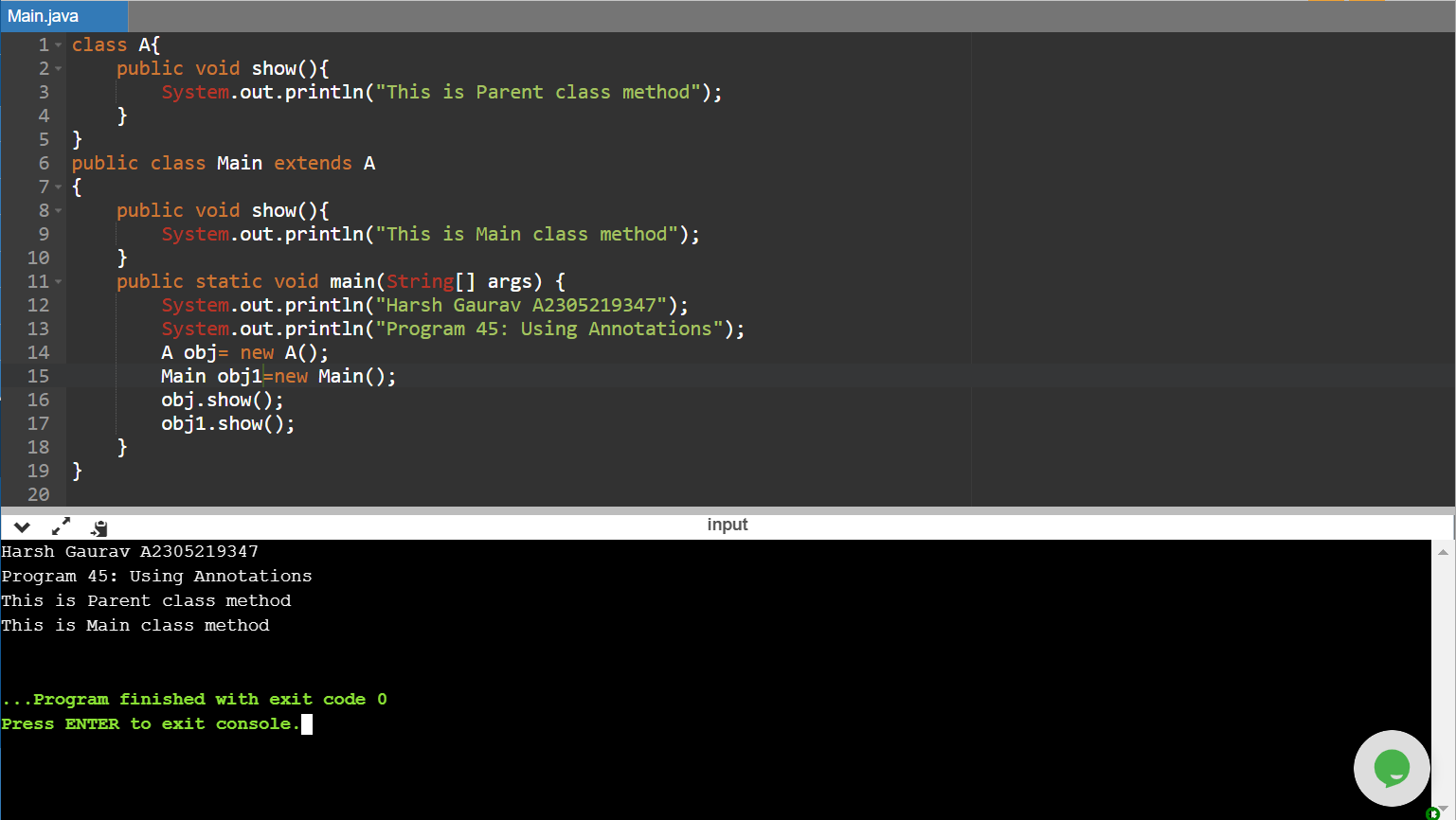
s1.showdata();

s2.showdata();

}

}

Q45. Write a program to show use of annotations.



class A{

public void show(){

System.out.println("This is Parent class method");

}

}

public class Main extends A

{

public void show(){

System.out.println("This is Main class method");

}

public static void main(String[] args) {

System.out.println("Harsh Gaurav A2305219347");

System.out.println("Program 45: Using Annotations");

A obj= new A();

Main obj1=new Main();

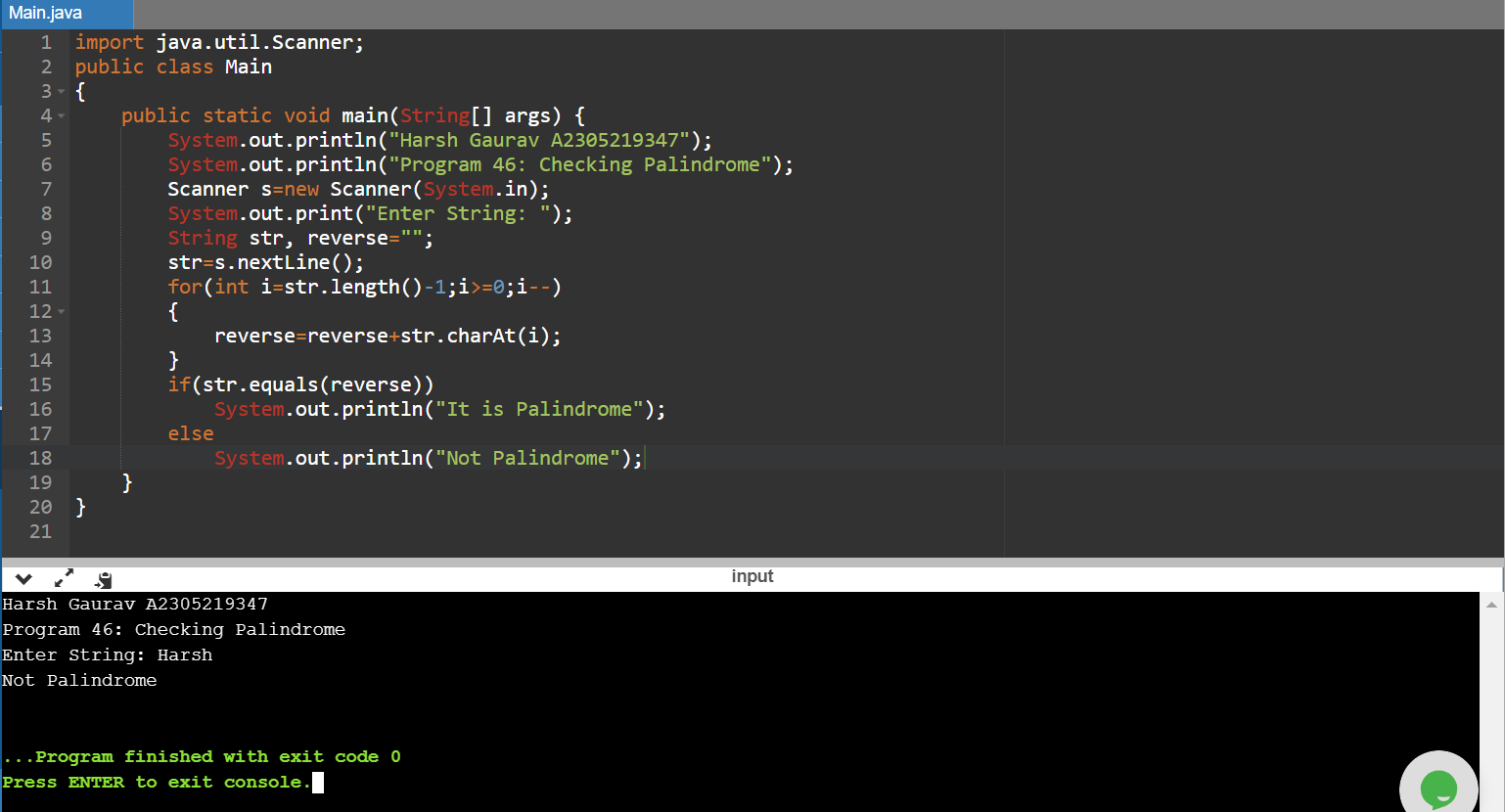
obj.show();

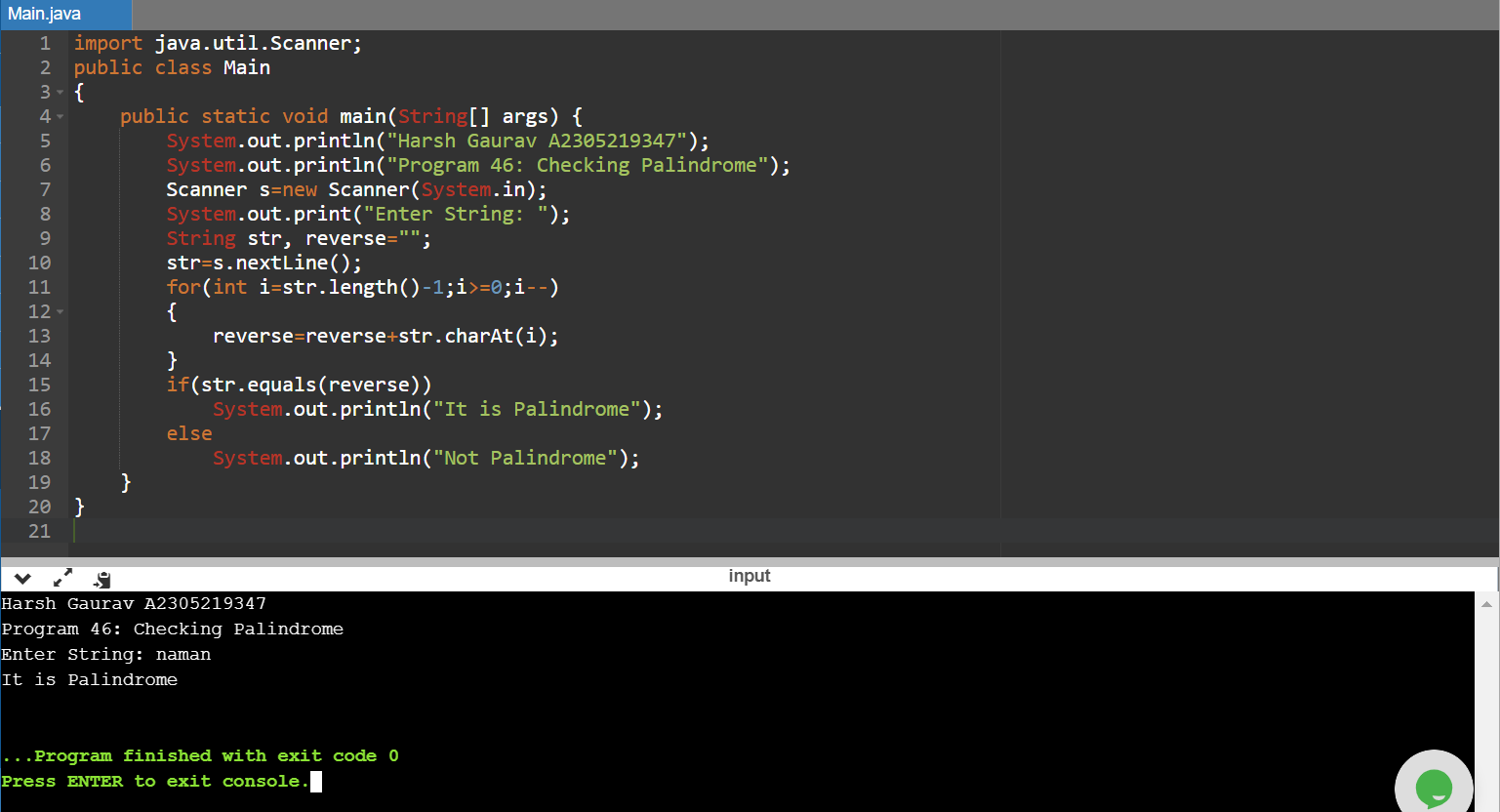
obj1.show();

}

}

Q46. Write a program to check whether a string is palindrome or not.





import java.util.Scanner;

public class Main

{

public static void main(String[] args) {

System.out.println("Harsh Gaurav A2305219347");

System.out.println("Program 46: Checking Palindrome");

Scanner s=new Scanner(System.in);

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

String str, reverse="";

str=s.nextLine();

for(int i=str.length()-1;i>=0;i--)

{

reverse=reverse+str.charAt(i);

}

if(str.equals(reverse))

System.out.println("It is Palindrome");

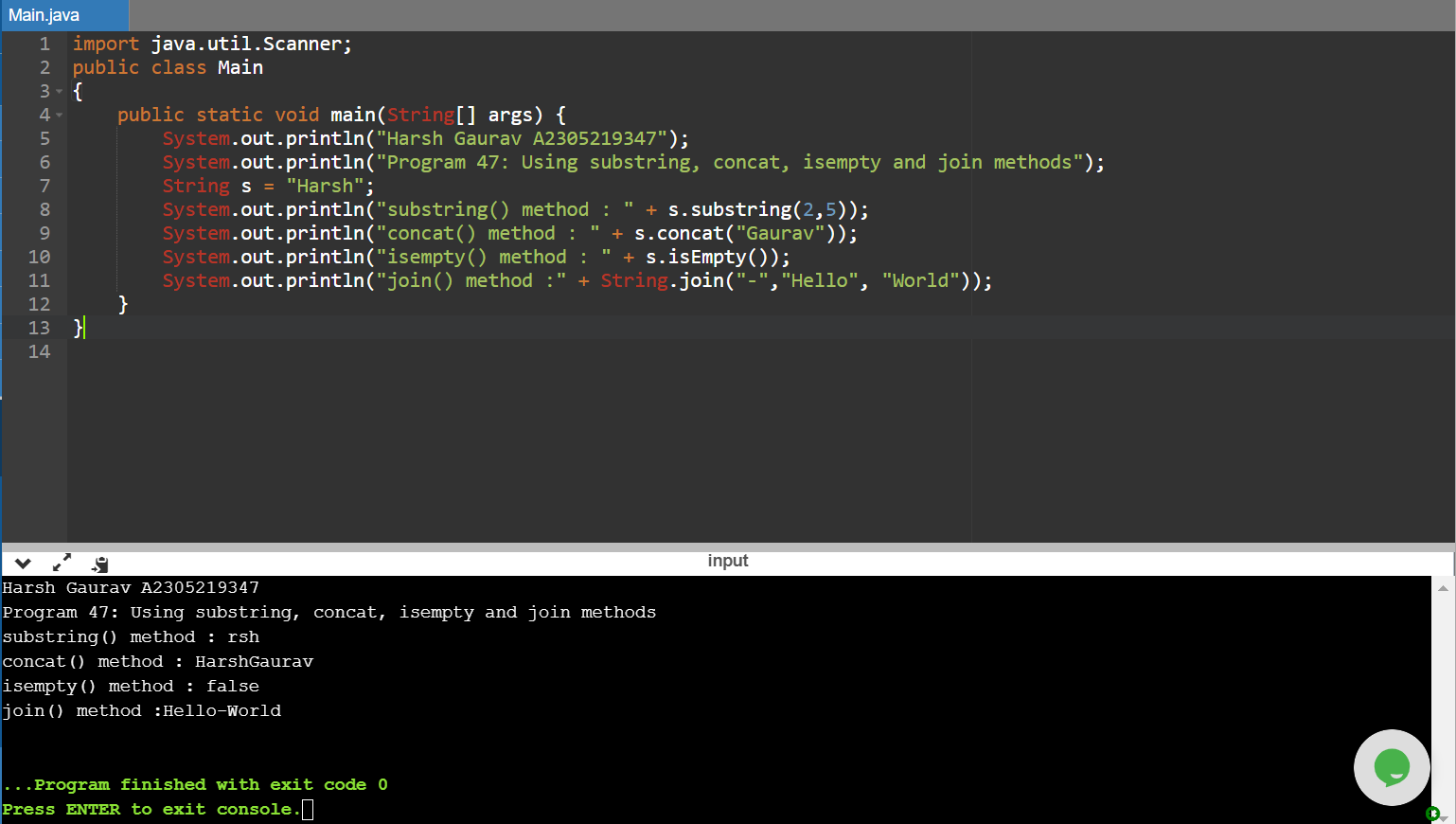
else

System.out.println("Not Palindrome");

}

}

Q47. Write a program to show use of string methods like substring, concat, isempty, join etc.



import java.util.Scanner;

public class Main

{

public static void main(String[] args) {

System.out.println("Harsh Gaurav A2305219347");

System.out.println("Program 47: Using substring, concat, isempty and join methods");

String s = "Harsh";

System.out.println("substring() method : " + s.substring(2,5));

System.out.println("concat() method : " + s.concat("Gaurav"));

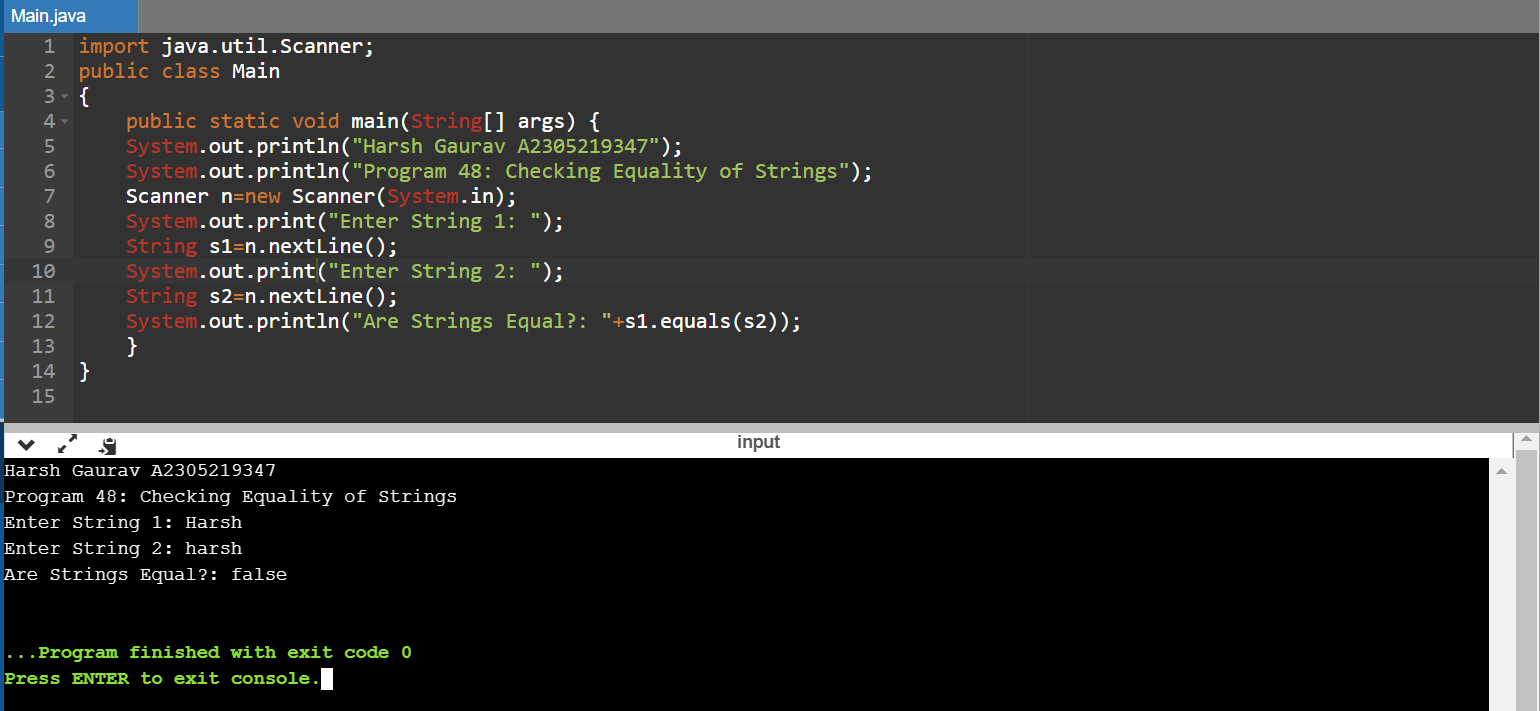
System.out.println("isempty() method : " + s.isEmpty());

System.out.println("join() method :" + String.join("-","Hello", "World"));

}

}

Q48. Write a program to check whether two strings are equal or not.



import java.util.Scanner;

public class Main

{

public static void main(String[] args) {

System.out.println("Harsh Gaurav A2305219347");

System.out.println("Program 48: Checking Equality of Strings");

Scanner n=new Scanner(System.in);

System.out.print("Enter String 1: ");

String s1=n.nextLine();

System.out.print("Enter String 2: ");

String s2=n.nextLine();

System.out.println("Are Strings Equal?: "+s1.equals(s2));

}

}

Q49. Write a program to show use of string methods like indexof, split, trim etc.



import java.util.Scanner;

public class Main

{

public static void main(String[] args) {

System.out.println("Harsh Gaurav A2305219347");

System.out.println("Program 49: Using indexof, split, trim methods");

String s = "How much you learn, it will be never enough";

System.out.println("indexof() method : " +s.indexOf("it"));

System.out.println("split() method : ");

String str = "Hello-World-Java";

System.out.println("Original String : " + str);

String[] split = str.split("-");

for (String w : split){

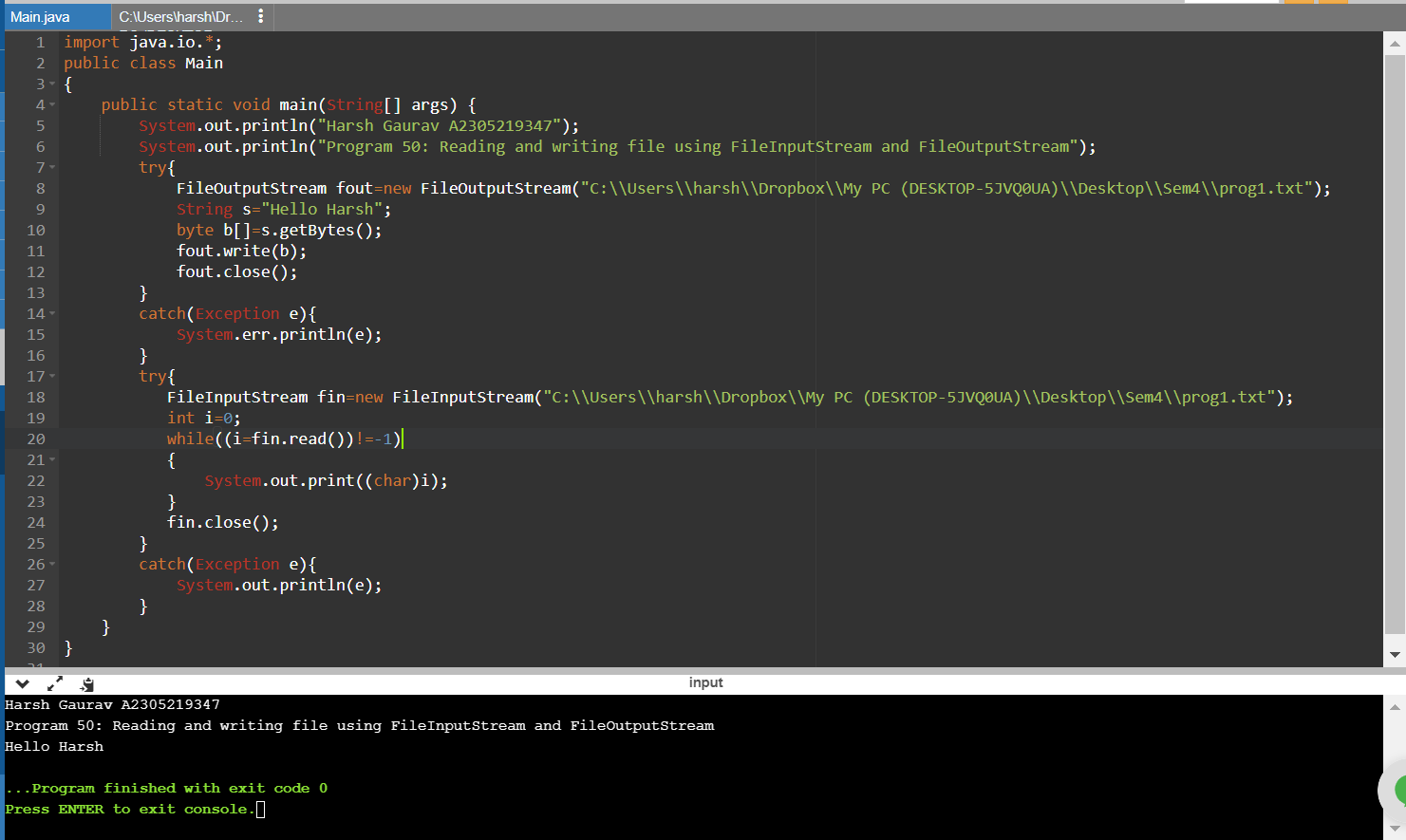
System.out.println("\t" + w);

}

System.out.println("trim() method : " + s.trim()); }}

01-03-2021

Q50. Write a program to create a file, write data into it. Then read the data from the same file and display on screen. Use FileInputStream and FileOutputStream.





import java.io.\*;

public class Main

{

public static void main(String[] args) {

System.out.println("Harsh Gaurav A2305219347");

System.out.println("Program 50: Reading and writing file using FileInputStream and FileOutputStream");

try{

FileOutputStream fout=new FileOutputStream("C:\\Users\\harsh\\Dropbox\\My PC (DESKTOP-5JVQ0UA)\\Desktop\\Sem4\\prog1.txt");

String s="Hello Harsh";

byte b[]=s.getBytes();

fout.write(b);

fout.close();

}

catch(Exception e){

System.err.println(e);

}

try{

FileInputStream fin=new FileInputStream("C:\\Users\\harsh\\Dropbox\\My PC (DESKTOP-5JVQ0UA)\\Desktop\\Sem4\\prog1.txt");

int i=0;

while((i=fin.read())!=-1)

{

System.out.print((char)i);

}

fin.close();

}

catch(Exception e){

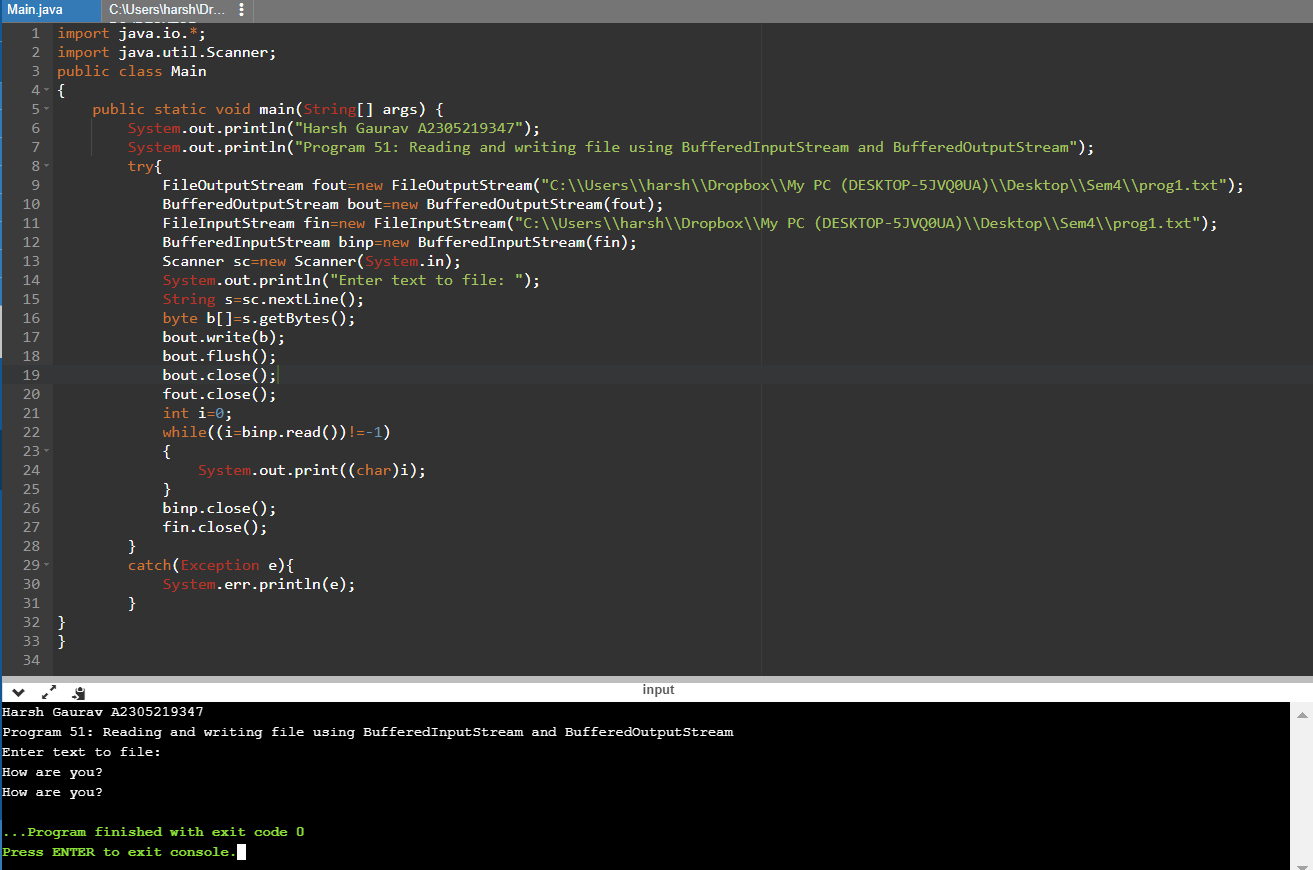
System.out.println(e);

}

}

}

Q51. Write a program to create a file, write data into it. Then read the data from the same file and display on screen. Use BufferedInputStream and BufferedOutputStream.



import java.io.\*;

import java.util.Scanner;

public class Main

{

public static void main(String[] args) {

System.out.println("Harsh Gaurav A2305219347");

System.out.println("Program 51: Reading and writing file using BufferedInputStream and BufferedOutputStream");

try{

FileOutputStream fout=new FileOutputStream("C:\\Users\\harsh\\Dropbox\\My PC (DESKTOP-5JVQ0UA)\\Desktop\\Sem4\\prog1.txt");

BufferedOutputStream bout=new BufferedOutputStream(fout);

FileInputStream fin=new FileInputStream("C:\\Users\\harsh\\Dropbox\\My PC (DESKTOP-5JVQ0UA)\\Desktop\\Sem4\\prog1.txt");

BufferedInputStream binp=new BufferedInputStream(fin);

Scanner sc=new Scanner(System.in);

System.out.println("Enter text to file: ");

String s=sc.nextLine();

byte b[]=s.getBytes();

bout.write(b);

bout.flush();

bout.close();

fout.close();

int i=0;

while((i=binp.read())!=-1)

{

System.out.print((char)i);

}

binp.close();

fin.close();

}

catch(Exception e){

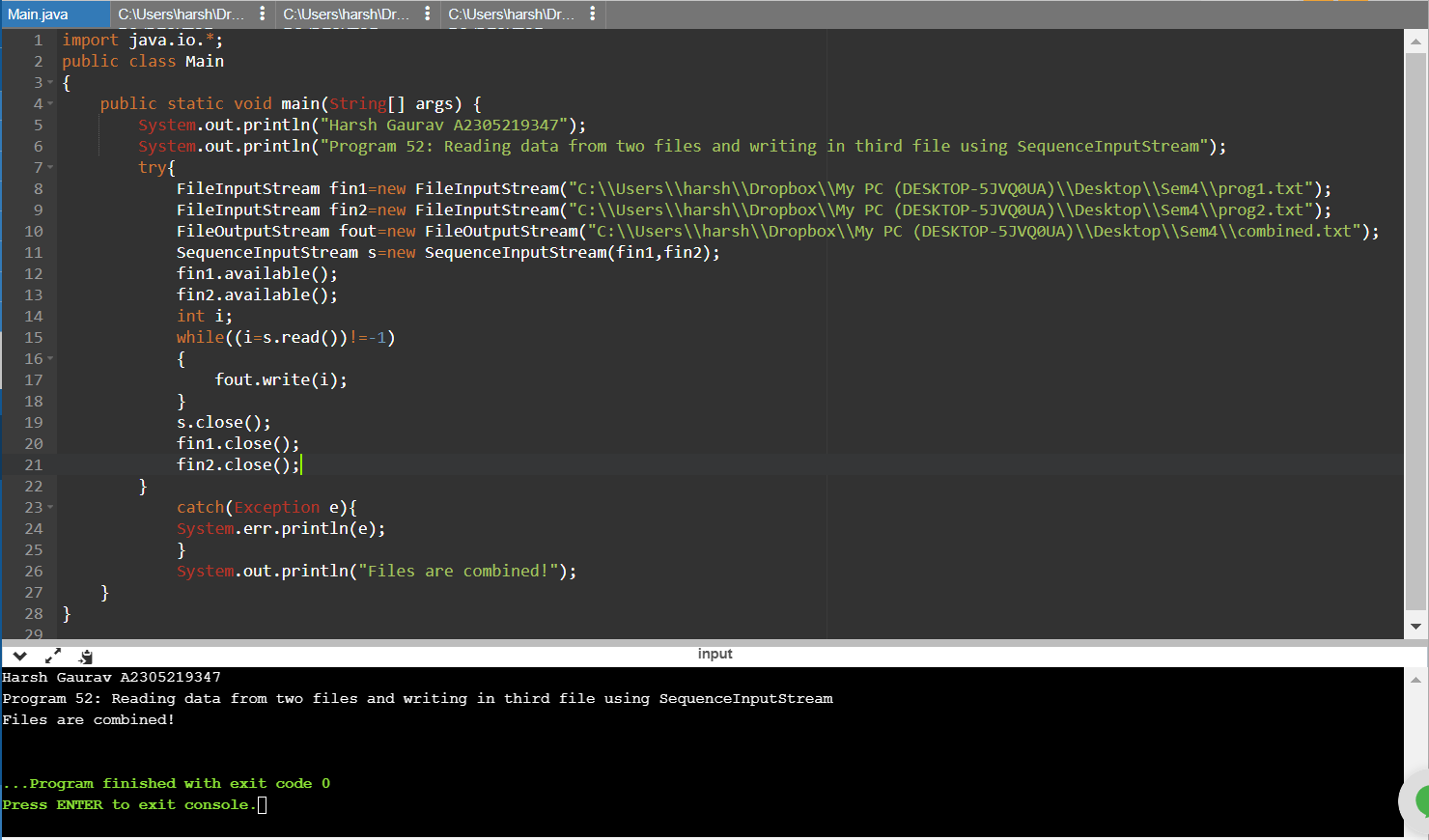
System.err.println(e);

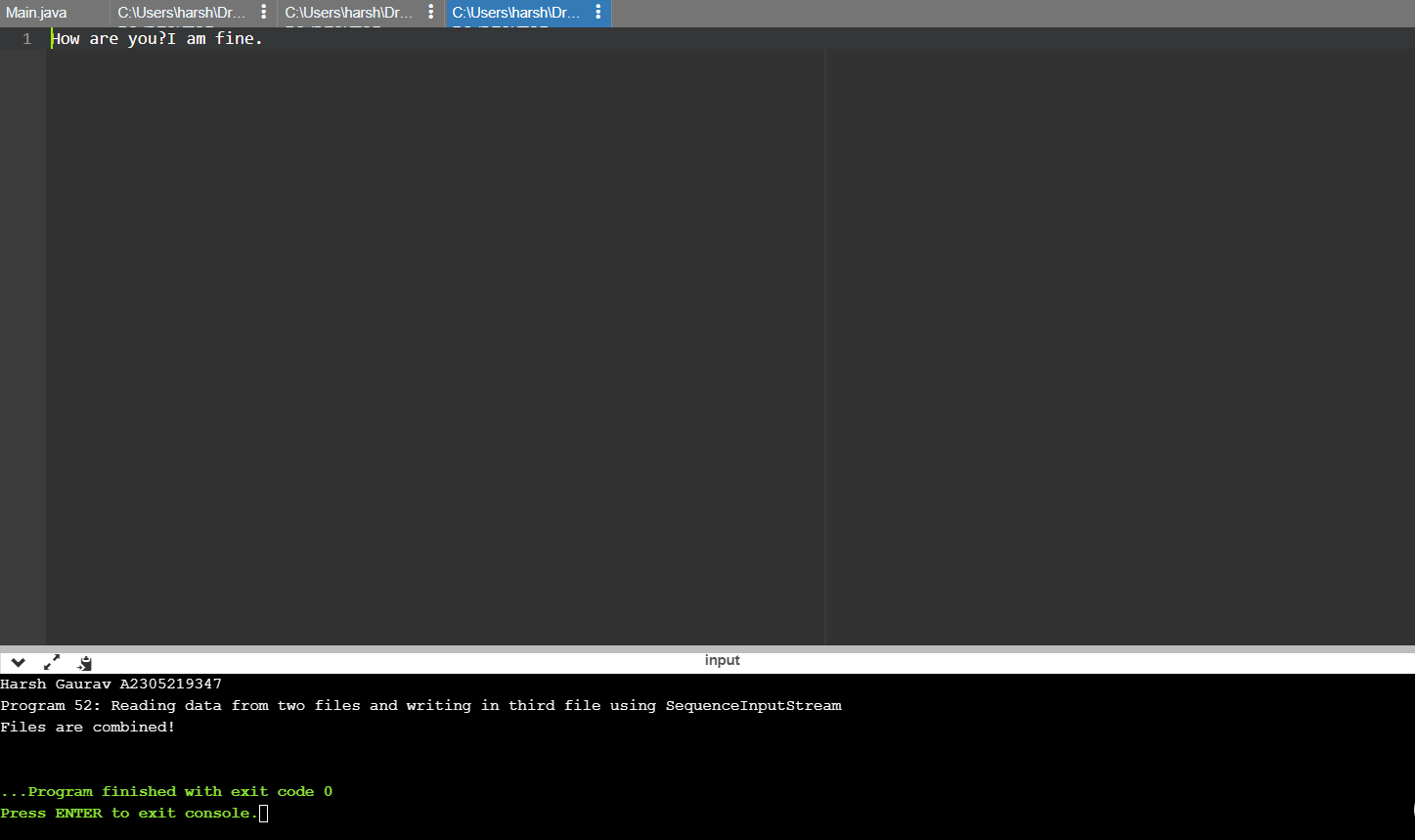
}

}

}

Q52. Using SequenceInputStream, write a program to read data from two files and write into third file.





import java.io.\*;

public class Main

{

public static void main(String[] args) {

System.out.println("Harsh Gaurav A2305219347");

System.out.println("Program 52: Reading data from two files and writing in third file using SequenceInputStream");

try{

FileInputStream fin1=new FileInputStream("C:\\Users\\harsh\\Dropbox\\My PC (DESKTOP-5JVQ0UA)\\Desktop\\Sem4\\prog1.txt");

FileInputStream fin2=new FileInputStream("C:\\Users\\harsh\\Dropbox\\My PC (DESKTOP-5JVQ0UA)\\Desktop\\Sem4\\prog2.txt");

FileOutputStream fout=new FileOutputStream("C:\\Users\\harsh\\Dropbox\\My PC (DESKTOP-5JVQ0UA)\\Desktop\\Sem4\\combined.txt");

SequenceInputStream s=new SequenceInputStream(fin1,fin2);

fin1.available();

fin2.available();

int i;

while((i=s.read())!=-1)

{

fout.write(i);

}

s.close();

fin1.close();

fin2.close();

}

catch(Exception e){

System.err.println(e);

}

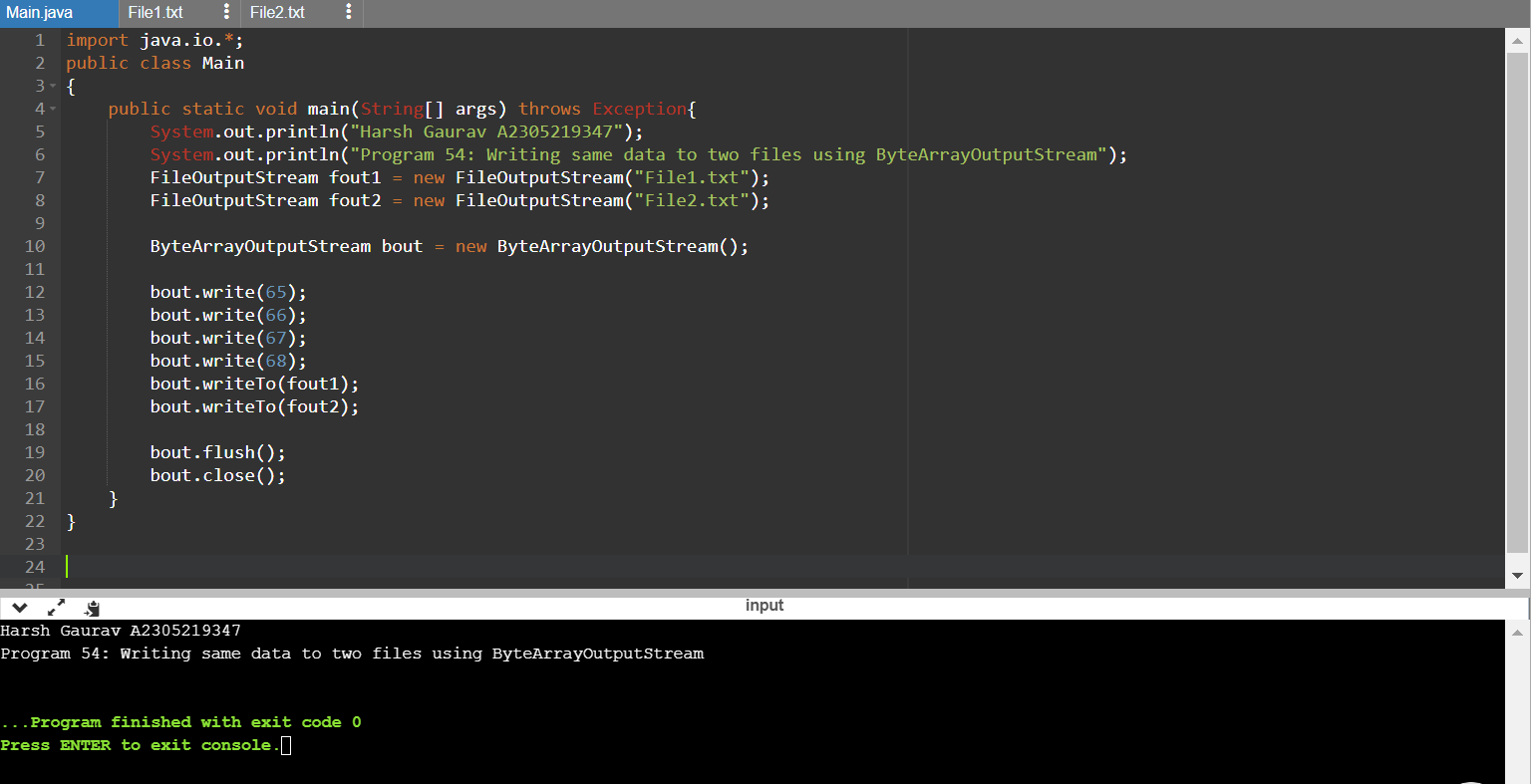
System.out.println("Files are combined!");

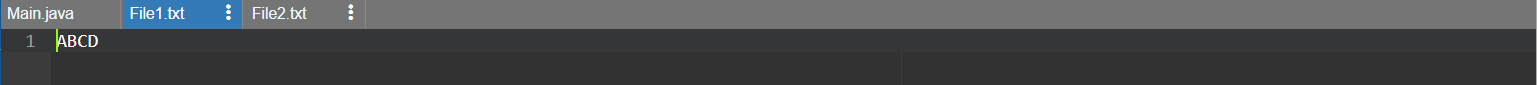
}

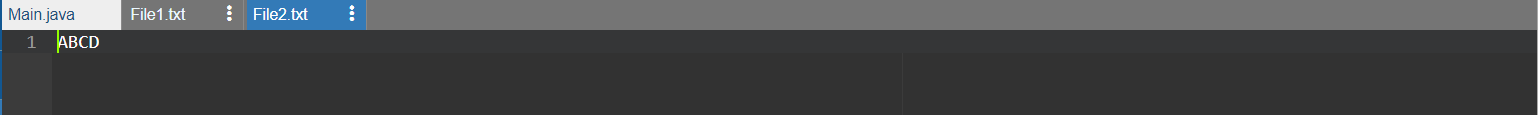
}

08-03-2021

Q53. Using ByteArrayOutputStream, write a program in Java to write same data to two files.







import java.io.\*;

public class Main

{

public static void main(String[] args) throws Exception{

System.out.println("Harsh Gaurav A2305219347");

System.out.println("Program 53: Writing same data to two files using ByteArrayOutputStream");

FileOutputStream fout1 = new FileOutputStream("File1.txt");

FileOutputStream fout2 = new FileOutputStream("File2.txt");

ByteArrayOutputStream bout = new ByteArrayOutputStream();

bout.write(65);

bout.write(66);

bout.write(67);

bout.write(68);

bout.writeTo(fout1);

bout.writeTo(fout2);

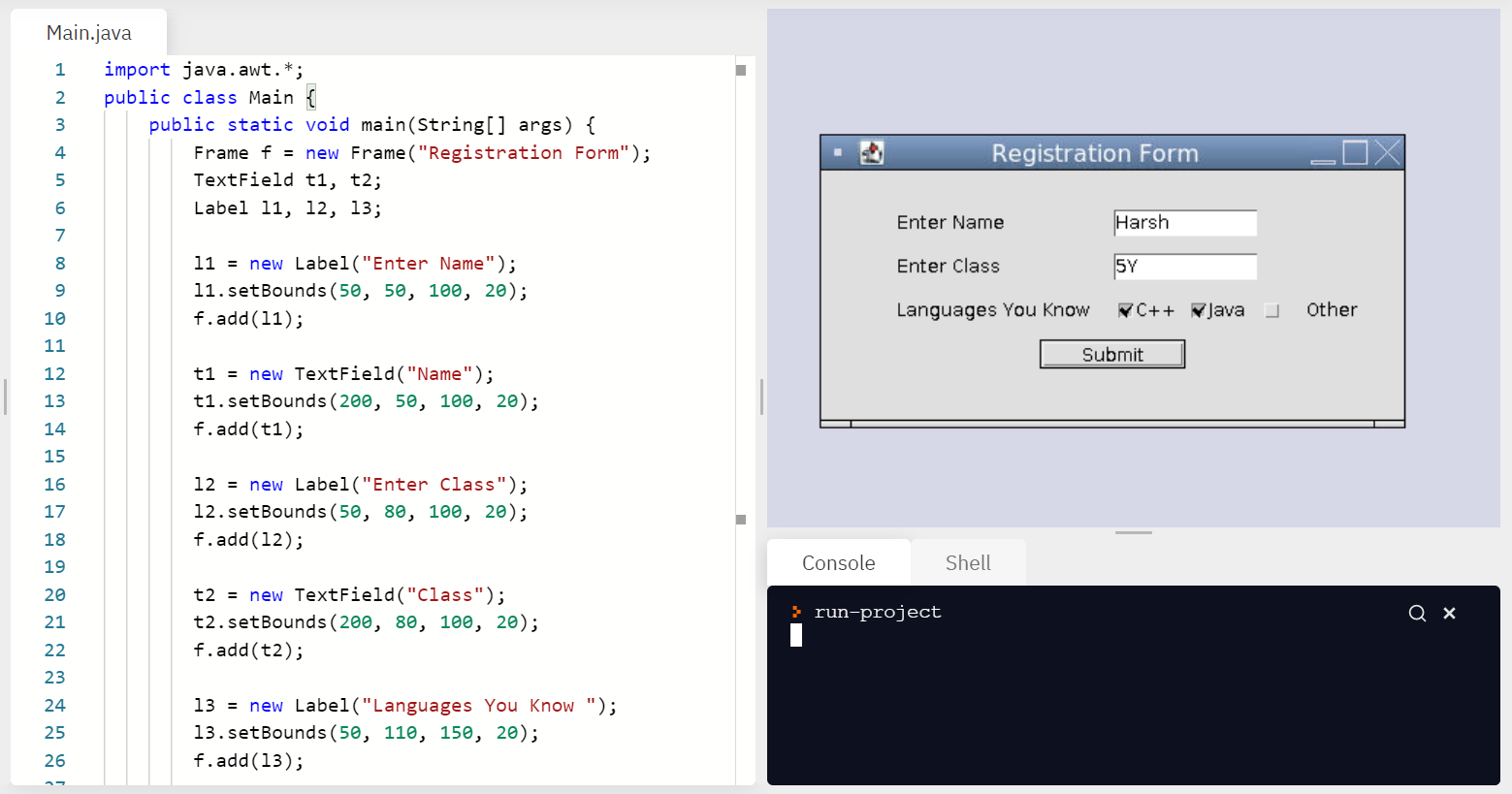
bout.flush();

bout.close();

}

}

Q54. Design a registration form in java using awt. Use components like textfield, button, checkbox etc.



import java.awt.\*;

public class Main {

public static void main(String[] args) {

Frame f = new Frame("Registration Form");

TextField t1, t2;

Label l1, l2, l3;

l1 = new Label("Enter Name");

l1.setBounds(50, 50, 100, 20);

f.add(l1);

t1 = new TextField("Name");

t1.setBounds(200, 50, 100, 20);

f.add(t1);

l2 = new Label("Enter Class");

l2.setBounds(50, 80, 100, 20);

f.add(l2);

t2 = new TextField("Class");

t2.setBounds(200, 80, 100, 20);

f.add(t2);

l3 = new Label("Languages You Know ");

l3.setBounds(50, 110, 150, 20);

f.add(l3);

Checkbox c1 = new Checkbox("C++");

c1.setBounds(200, 110, 50, 20);

f.add(c1);

Checkbox c2 = new Checkbox("Java");

c2.setBounds(250, 110, 50, 20);

f.add(c2);

Checkbox c3 = new Checkbox("Other");

c3.setBounds(300, 110, 100, 20);

f.add(c3);

Button b = new Button("Submit");

b.setBounds(150, 140, 100, 20);

f.add(b);

f.setSize(400, 200);

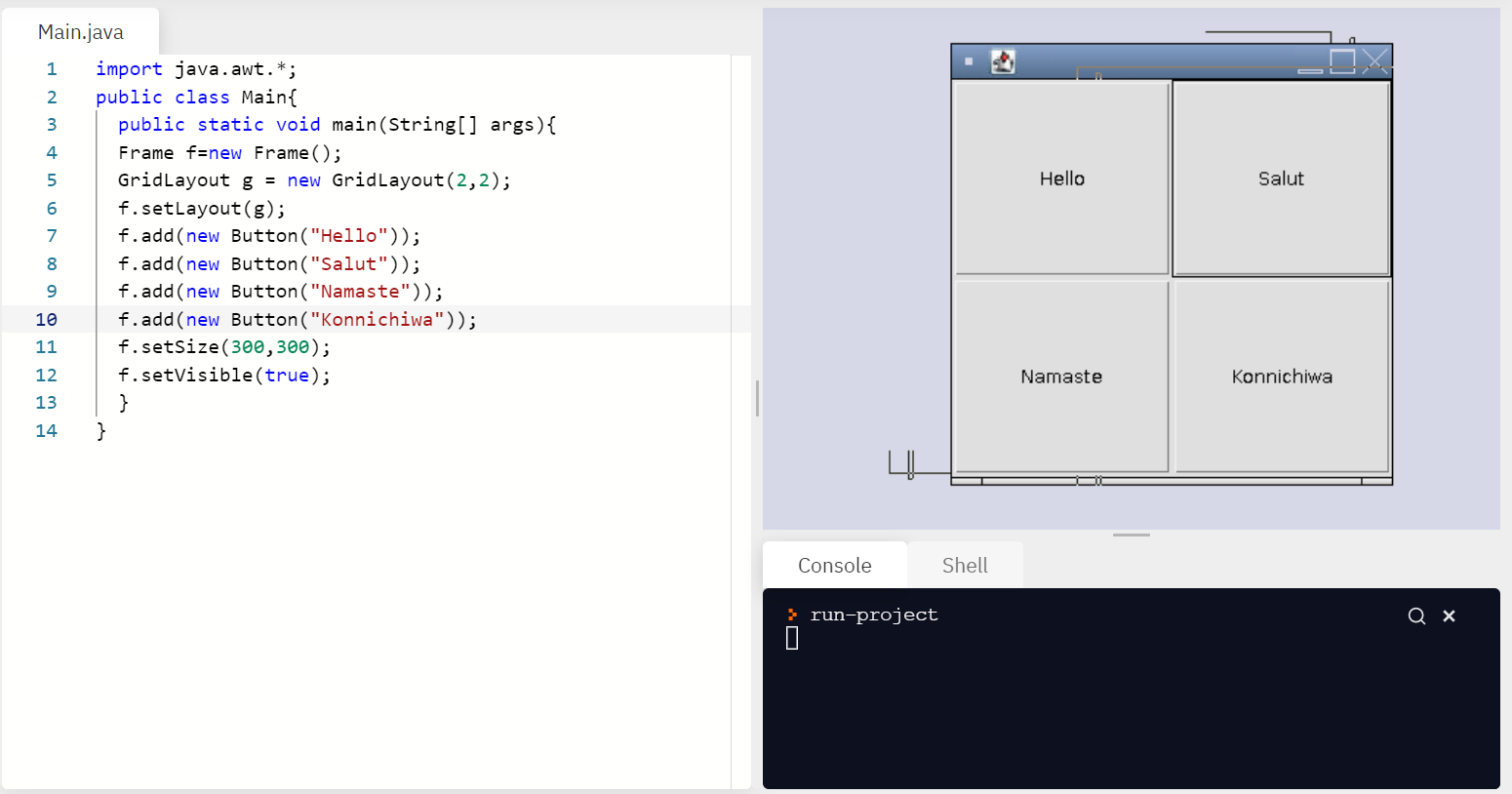
f.setLayout(null);

f.setVisible(true);

}

}

Q55. Use GridLayoutManager using awt to place components like textfield, button, checkbox etc.



import java.awt.\*;

public class Main{

public static void main(String[] args){

Frame f=new Frame();

GridLayout g = new GridLayout(2,2);

f.setLayout(g);

f.add(new Button("Hello"));

f.add(new Button("Salut"));

f.add(new Button("Namaste"));

f.add(new Button("Konnichiwa"));

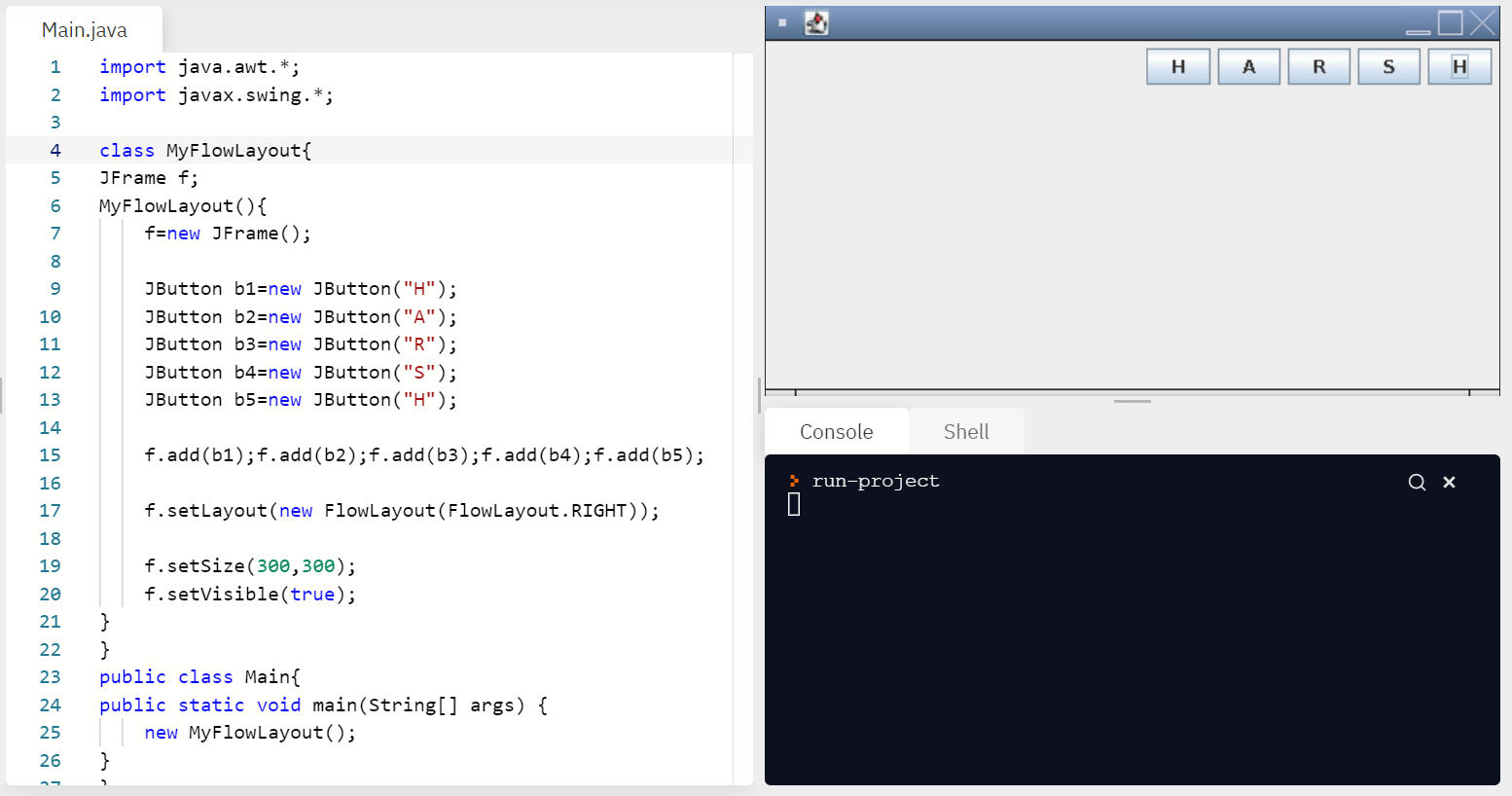
f.setSize(300,300);

f.setVisible(true);

}

}

Q56. Use FlowLayoutManager using awt to place components like textfield, button, checkbox etc.



import java.awt.\*;

import javax.swing.\*;

class MyFlowLayout{

JFrame f;

MyFlowLayout(){

f=new JFrame();

JButton b1=new JButton("H");

JButton b2=new JButton("A");

JButton b3=new JButton("R");

JButton b4=new JButton("S");

JButton b5=new JButton("H");

f.add(b1);f.add(b2);f.add(b3);f.add(b4);f.add(b5);

f.setLayout(new FlowLayout(FlowLayout.RIGHT));

f.setSize(300,300);

f.setVisible(true);

}

}

public class Main{

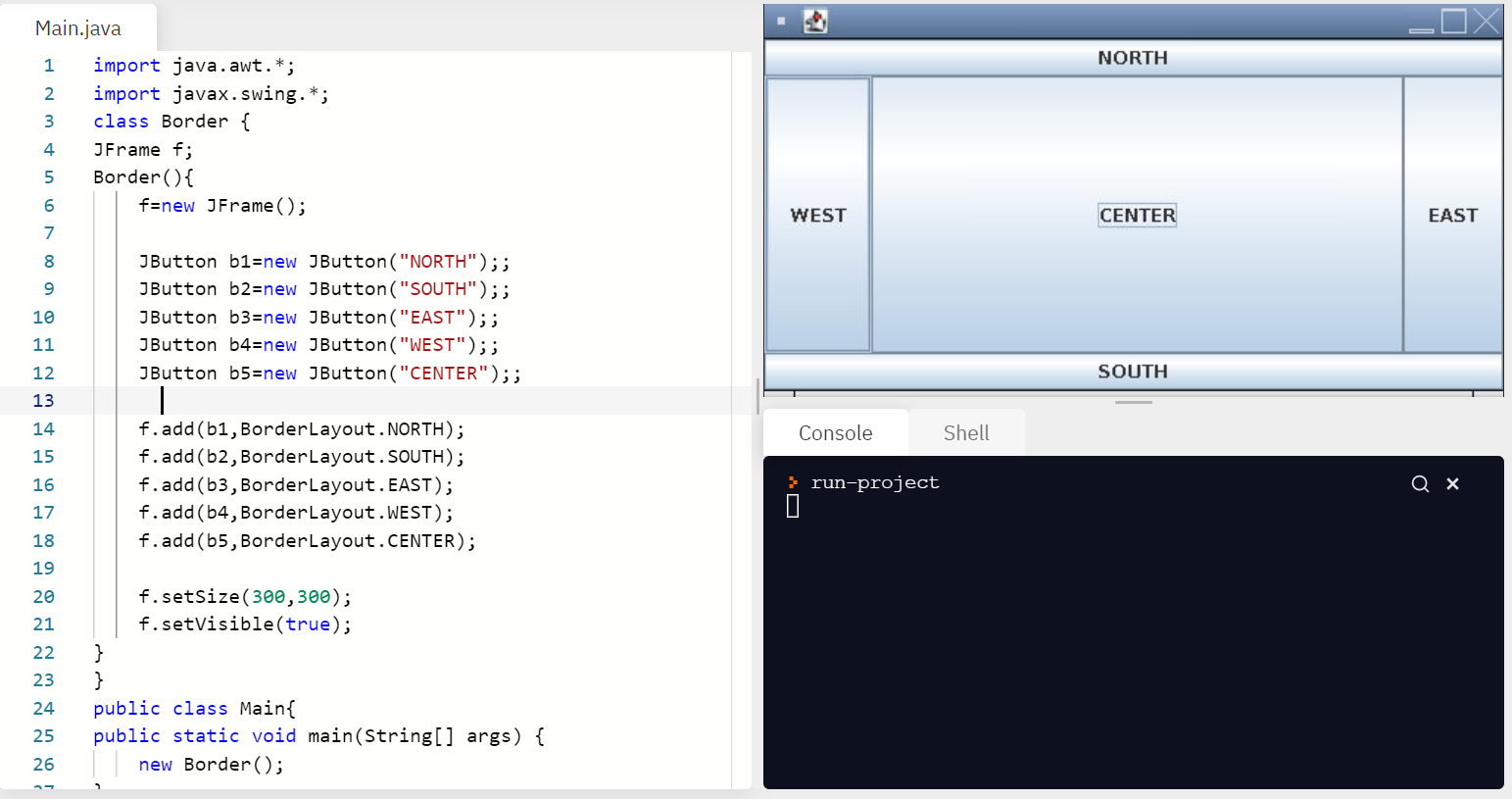
public static void main(String[] args) {

new MyFlowLayout();

}

}

Q57. Use BorderLayoutManager using awt to place components like textfield, button, checkbox etc.



import java.awt.\*;

import javax.swing.\*;

class Border {

JFrame f;

Border(){

f=new JFrame();

JButton b1=new JButton("NORTH");;

JButton b2=new JButton("SOUTH");;

JButton b3=new JButton("EAST");;

JButton b4=new JButton("WEST");;

JButton b5=new JButton("CENTER");;

f.add(b1,BorderLayout.NORTH);

f.add(b2,BorderLayout.SOUTH);

f.add(b3,BorderLayout.EAST);

f.add(b4,BorderLayout.WEST);

f.add(b5,BorderLayout.CENTER);

f.setSize(300,300);

f.setVisible(true);

}

}

public class Main{

public static void main(String[] args) {

new Border();

}

}

15-03-2021

Q58. Create Labels (One, Two & Three), add Buttons (Yes, No, Undecided) when user click any button show message regarding user click, add Checkboxes(Windows 98/XP, WindowsNT/2000) when user chose any checkbox show message regarding user choice & add text boxes (name, password) and text on these textboxes should be displayed on Panel.

import java.awt.\*;

import java.awt.event.\*;

import java.applet.\*;

/\*

<applet code="CheckboxDemo" width=250 height=200>

</applet>

\*/

public class CheckboxDemo extends Applet implements ItemListener

{

String msg = "";

Checkbox Win98, winNT, solaris, mac;

public void init()

{

Win98 = new Checkbox("Windows 98/XP", null, true);

winNT = new Checkbox("Windows NT/2000");

solaris = new Checkbox("Solaris");

mac = new Checkbox("MacOS");

add(Win98);

add(winNT);

add(solaris);

add(mac);

Win98.addItemListener(this);

winNT.addItemListener(this);

solaris.addItemListener(this);

mac.addItemListener(this);

}

public void itemStateChanged(ItemEvent ie)

{

repaint();

}

// Display current state of the check boxes.

public void paint(Graphics g)

{

msg = "Current state: ";

g.drawString(msg, 6, 80);

msg = " Windows 98/XP: " + Win98.getState();

g.drawString(msg, 6, 100);

msg = " Windows NT/2000: " + winNT.getState();

g.drawString(msg, 6, 120);

msg = " Solaris: " + solaris.getState();

g.drawString(msg, 6, 140);

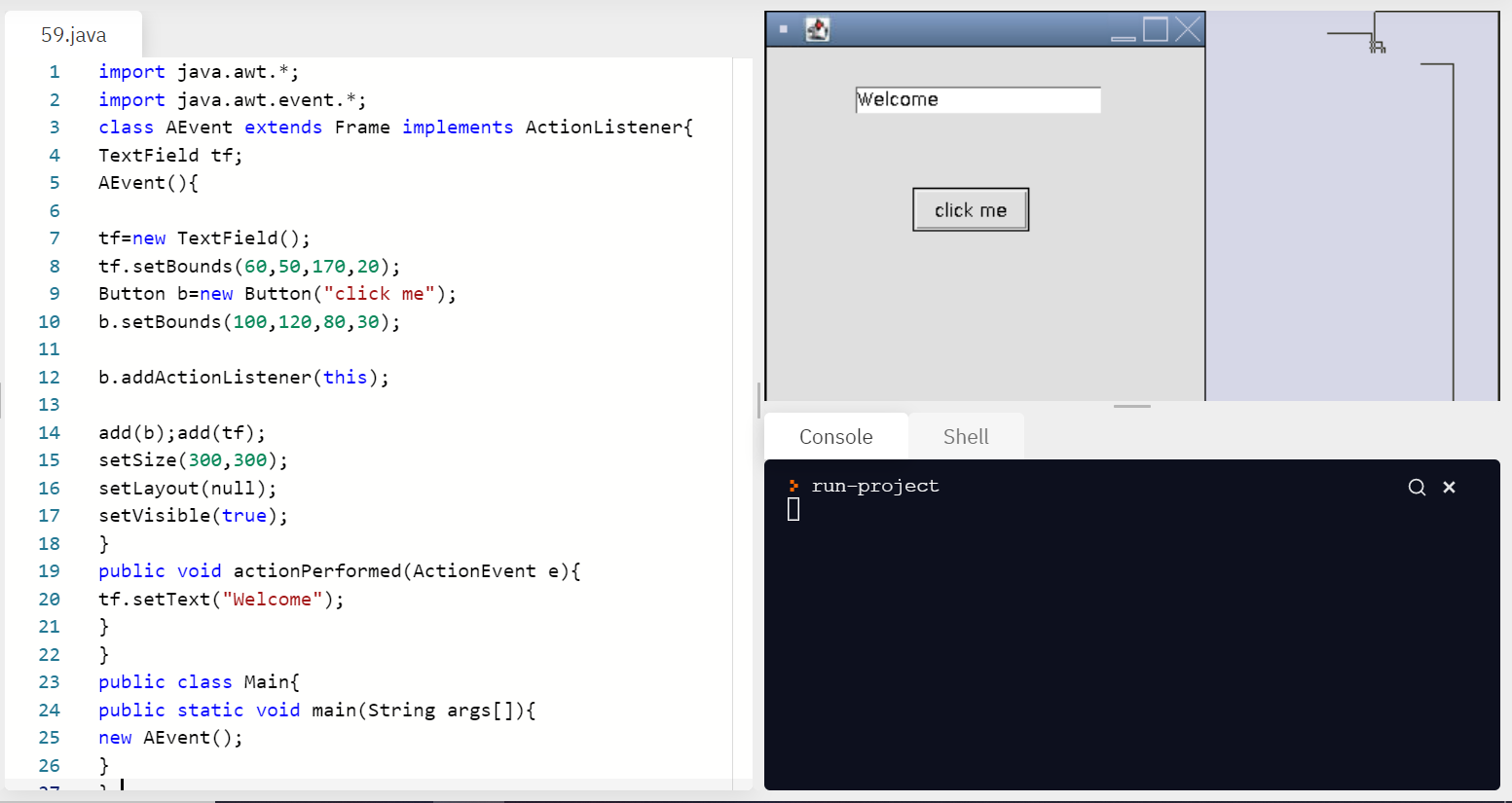
msg = " MacOS: " + mac.getState();

g.drawString(msg, 6, 160);

}

}

Q59. Write a program in Java to implement event handling code. Implement the event handling code within the class.



import java.awt.\*;

import java.awt.event.\*;

class AEvent extends Frame implements ActionListener{

TextField tf;

AEvent(){

tf=new TextField();

tf.setBounds(60,50,170,20);

Button b=new Button("click me");

b.setBounds(100,120,80,30);

b.addActionListener(this);

add(b);add(tf);

setSize(300,300);

setLayout(null);

setVisible(true);

}

public void actionPerformed(ActionEvent e){

tf.setText("Welcome");

}

}

public class Main{

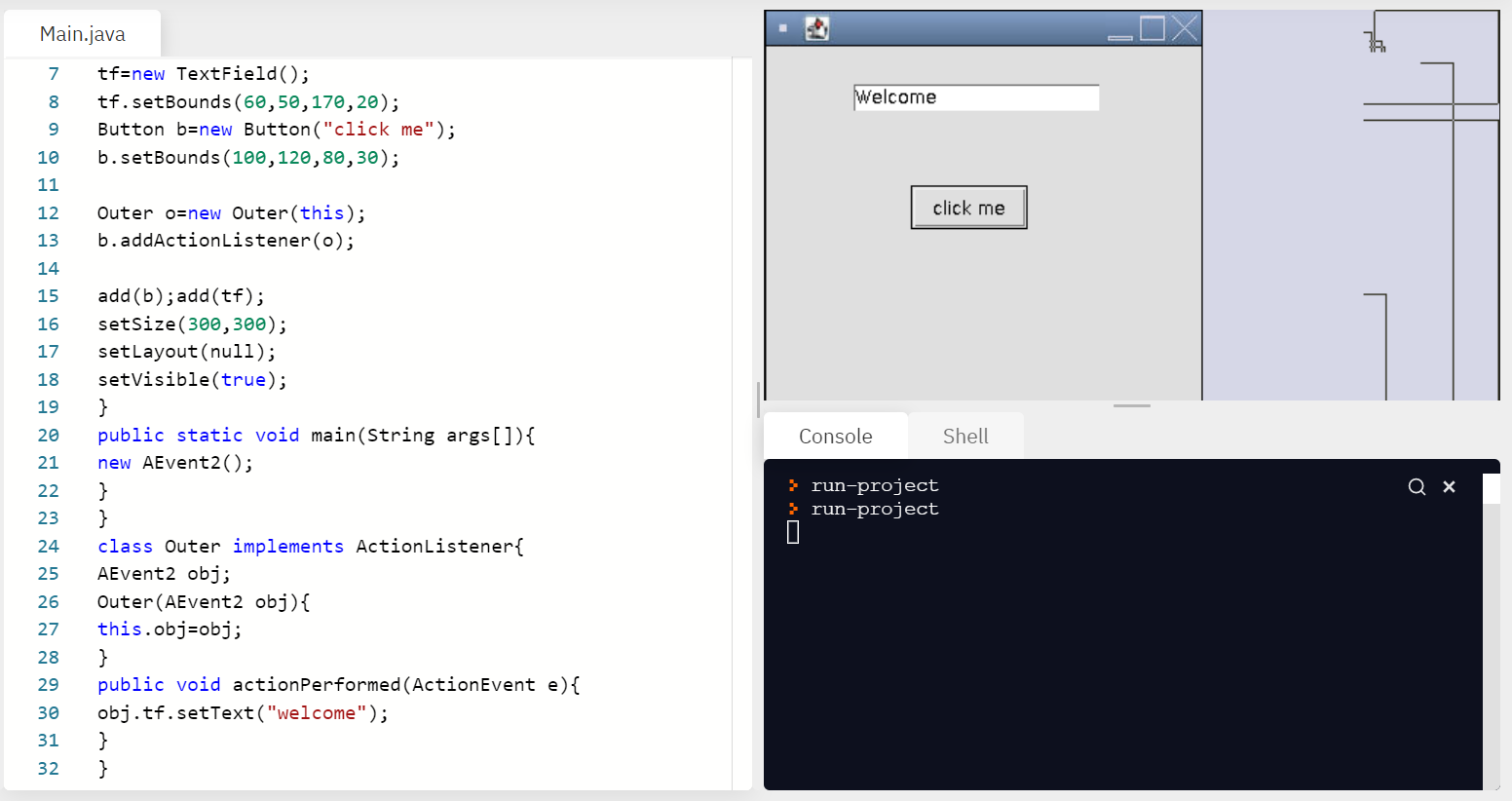
public static void main(String args[]){

new AEvent();

}

}

Q60. Write a program in java to implement event handling code. Implement the event handling code using outer class.



import java.awt.\*;

import java.awt.event.\*;

class AEvent2 extends Frame{

TextField tf;

AEvent2(){

tf=new TextField();

tf.setBounds(60,50,170,20);

Button b=new Button("click me");

b.setBounds(100,120,80,30);

Outer o=new Outer(this);

b.addActionListener(o);

add(b);add(tf);

setSize(300,300);

setLayout(null);

setVisible(true);

}

public static void main(String args[]){

new AEvent2();

}

}

class Outer implements ActionListener{

AEvent2 obj;

Outer(AEvent2 obj){

this.obj=obj;

}

public void actionPerformed(ActionEvent e){

obj.tf.setText("welcome");

}

}