**LAB RECORD**

**BACHELOR OF TECHNOLOGY**

**B.Tech. CS&E -D, Semester ( 4 )**

**(Academic Session – 2022- 2023)**

|  |  |
| --- | --- |
| *Course Title* | *Java Programming* |
| *Course Code* | *IT201* |
| *Enrollment Number* | *A7605221184* |
| *Name of Student* | *Kunal Yadav* |
| *Date of Submission* | *\_\_/\_\_/\_\_\_* |
| *Signature of Student* |  |
| *Grade/Marks obtained* |  |
| *Faculty Name & signature* |  |

**Department of Computer Science & Engineering**

**Amity School of Engineering & Technology**

**Amity University, Lucknow Campus**

**CONTENT**

|  |  |  |  |
| --- | --- | --- | --- |
| S.No. | Topic | DATE | T.Sign. |
| 1 | Write a program in java to add two numbers. |  |  |
| 2 | Write a program in java to show increment and decrement operators. |  |  |
| 3 | Write a program in java to show logical and bitwise operators. |  |  |
| 4 | Write a program in java to extract last two digits of a number entered by the user. |  |  |
| 5 | Write a program in java to find the sum of the given series up to certain number of terms.  Series: 1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9 +… |  |  |
| 6 | Write a program in java to find the sum of the given series up to certain number of terms.  Series: 5 + 6 + 7/8 + 1/2 + 2/3 + 3/4 + 4/5 + … |  |  |
| 7 | Write a program in java to find the sum of the given series up to certain number of terms.  Series: 1 - 1/2 + 1/4 - 1/6 + 1/8 - 1/10 + … |  |  |
| 8 | Write a program in java to find the sum of the given series up to certain number of terms.  Series: 1 + 1/2! + 1/4! + 1/6! + 1/8! + 1/10! + … |  |  |
| 9 | Write a program in java to find the sum of the given series up to certain number of terms. [showing method calling within two classes]  Series: 1 + 1/2! + 1/4! + 1/6! + 1/8! + 1/10! + … |  |  |
| 10 | Write a program in java to find the sum of the given series up to certain number of terms. [showing use of static in classes]  Series: 1 + 1/2! + 1/4! + 1/6! + 1/8! + 1/10! + … |  |  |
| 11 | Write a program in java to calculate the area of a rectangle and a circle [using method overloading]. |  |  |
| 12 | Write a program in java to calculate the area of a rectangle and a circle [using constructor overloading]. |  |  |
| 13 | Write a program in java to check whether the given no is Armstrong Number. |  |  |
| 14 | Write a program in java to calculate the sum of square of the digits placed at the even positions of the number entered by the user. |  |  |
| 15 | Write a program in java to check whether the given input is Palindrome. |  |  |
| 16 | Write a program in java with a class named Box with the data members as length, breadth and height, and member functions as area() and volume(), also create another class containing main method to create objects and call the methods of the Box class. |  |  |
| 17 | Write a program in java to take input of 10 elements of an array and print the array also. |  |  |
| 18 | Write a program in java to calculate the sum of elements arrays of the elements. |  |  |
| 19 | Write a program in java for the following searching algorithms:   1. Linear Search 2. Binary Search |  |  |
| 20 | Write a program in java for the following sorting algorithms:   1. Bubble Sort 2. Insertion Sort |  |  |
| 21 | Write a program in java for the various types of inheritance. |  |  |
| 22 | Write a Program in java to show the implementation of threads |  |  |
| 23 | Write a program in java to show the implementation of try-catch-finally keywords. |  |  |
| 24 | Write a program in java to show the implementation of throw and throws. |  |  |
| 25 | Write a program in java AWT to design a login page. |  |  |

PROGRAM – 01

**Problem Statement:**

Write a program in java to add two numbers.

**Source Code:**

import java.util.Scanner;

public class First {

public static void main(String args[]) {

System.out.println("Enter the 2 no.:");

Scanner sc = new Scanner(System.in);

int a = sc.nextInt();

int b = sc.nextInt();

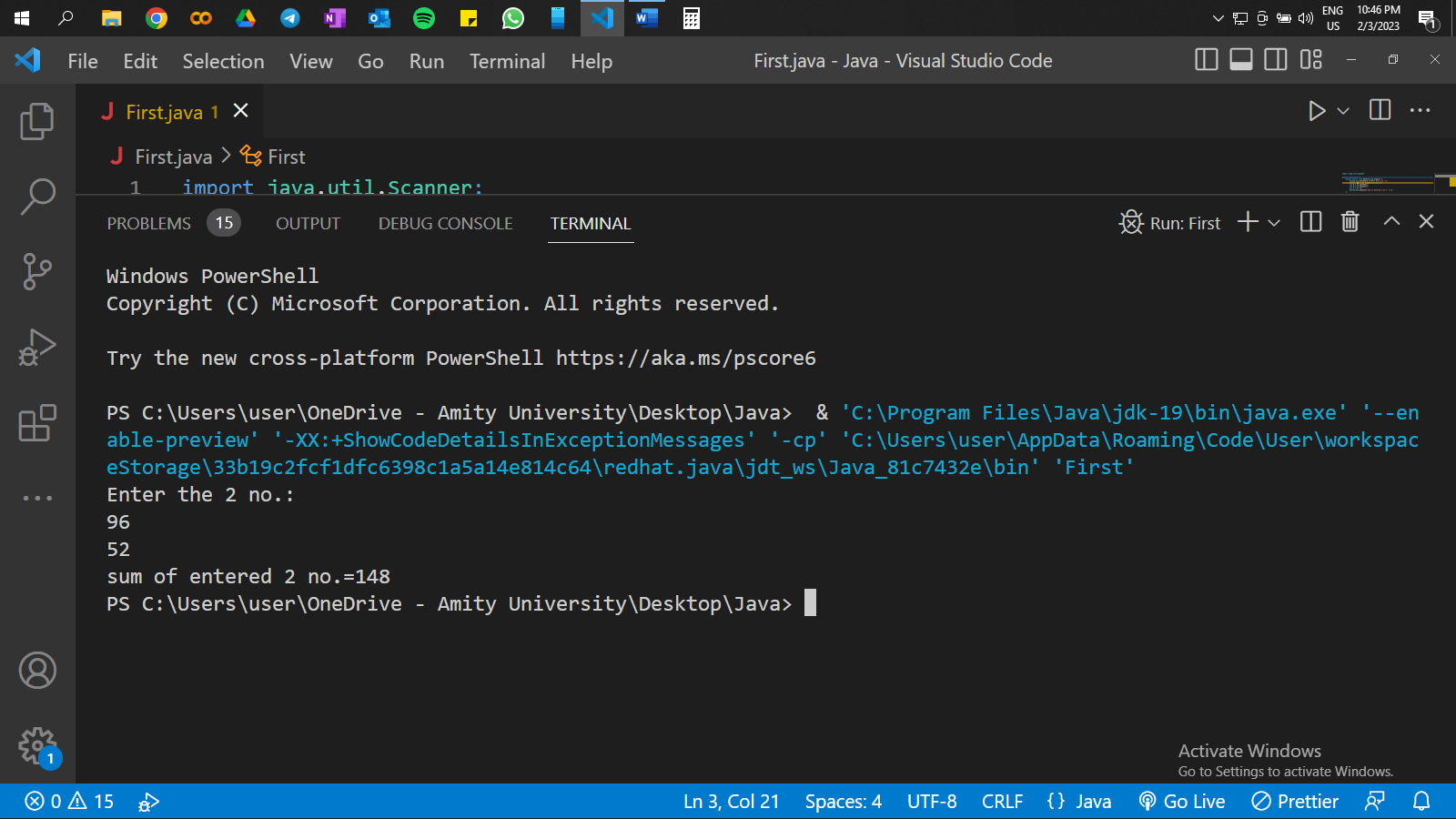
int c = a + b;

System.out.println("sum of entered 2 no.=" + c);

}

}

**Output:**



PROGRAM – 02

**Problem Statement:**

Write a program in java to show increment and decrement operators.

**Source Code:**

import java.util.\*;

public class IncerementAndDecrement {

public static void main(String args[]) {

System.out.println("Enter the no.:");

Scanner sc = new Scanner(System.in);

int b = sc.nextInt();

System.out.println("Result with pre-increment operator of the entered no. are:" + ++b);

System.out.println("Result after pre-increment operator of the entered no. are:" + b);

System.out.println("Result with post-increment operator of the entered no. are:" + b++);

System.out.println("Result after post-increment operator of the entered no. are:" + b);

System.out.println("Result with pre-decrement operator of the entered no. are:" + --b);

System.out.println("Result after pre-decrement operator of the entered no. are:" + b);

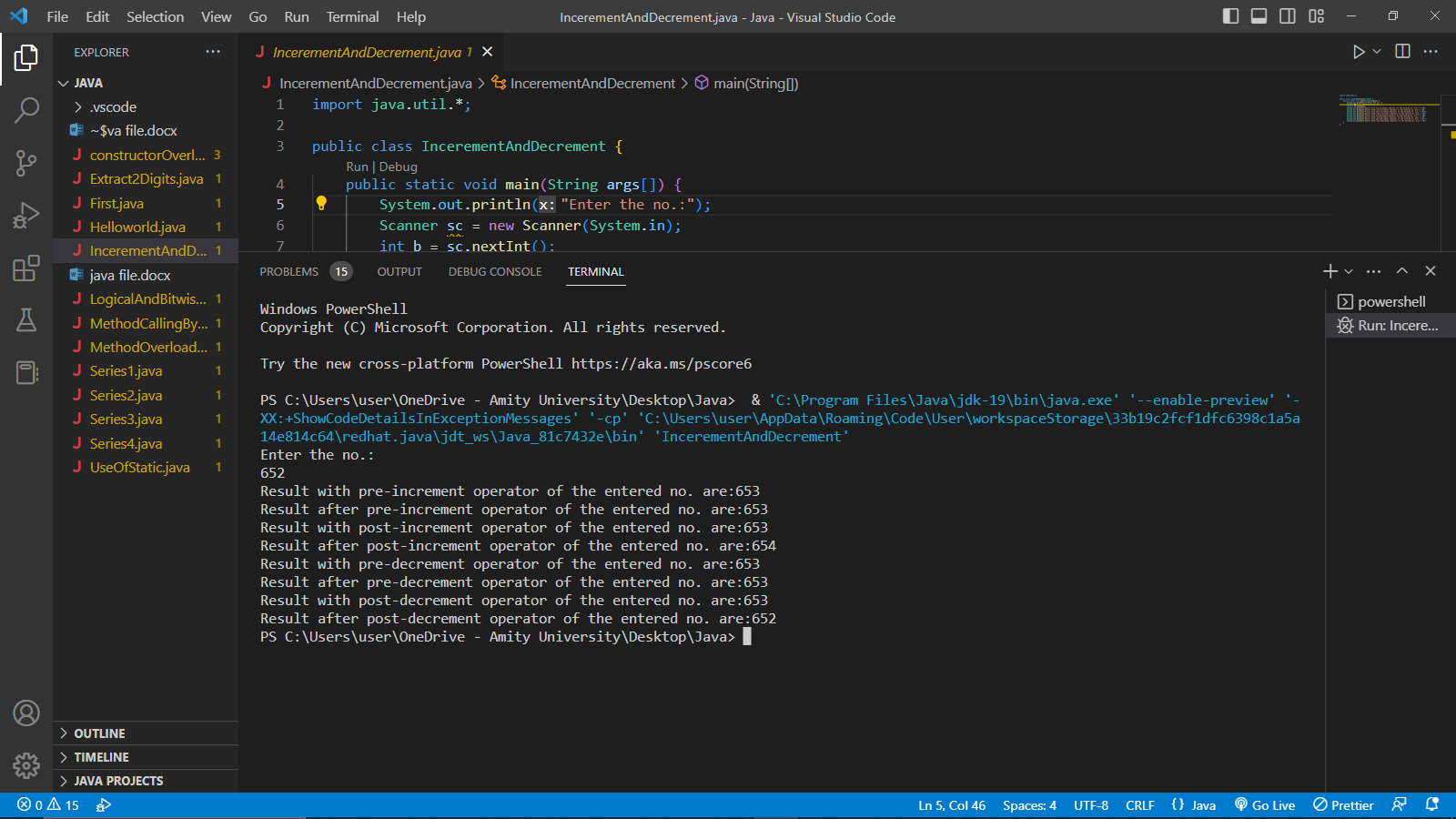
System.out.println("Result with post-decrement operator of the entered no. are:" + b--);

System.out.println("Result after post-decrement operator of the entered no. are:" + b);

}

}

**Output:**



PROGRAM – 03

**Problem Statement:**

Write a program in java to show logical and bitwise operators.

**Source Code:**

public class OperatorsExample {

public static void main(String[] args) {

int a = 6;

int b = 3;

// Logical Operators

boolean logicalAnd = (a > 5) && (b < 5);

boolean logicalOr = (a > 5) || (b < 5);

boolean logicalNot = !(a > 5);

System.out.println("Logical AND: " + logicalAnd); // prints false

System.out.println("Logical OR: " + logicalOr); // prints true

System.out.println("Logical NOT: " + logicalNot); // prints false

// Bitwise Operators

int bitwiseAnd = a & b;

int bitwiseOr = a | b;

int bitwiseXor = a ^ b;

int bitwiseComplement = ~a;

System.out.println("Bitwise AND: " + bitwiseAnd); // prints 2

System.out.println("Bitwise OR: " + bitwiseOr); // prints 7

System.out.println("Bitwise XOR: " + bitwiseXor); // prints 5

System.out.println("Bitwise Complement: " + bitwiseComplement); // prints -7

}

}

**Output:**

A screenshot of a computer

Description automatically generated with medium confidence

PROGRAM – 04

**Problem Statement:**

Write a program in java to extract last two digits of a number entered by the user.

**Source Code:**

import java.util.\*;

public class Extract2Digits {

public static void main(String args[]) {

System.out.println("Enter the no.:");

Scanner sc = new Scanner(System.in);

int a = sc.nextInt();

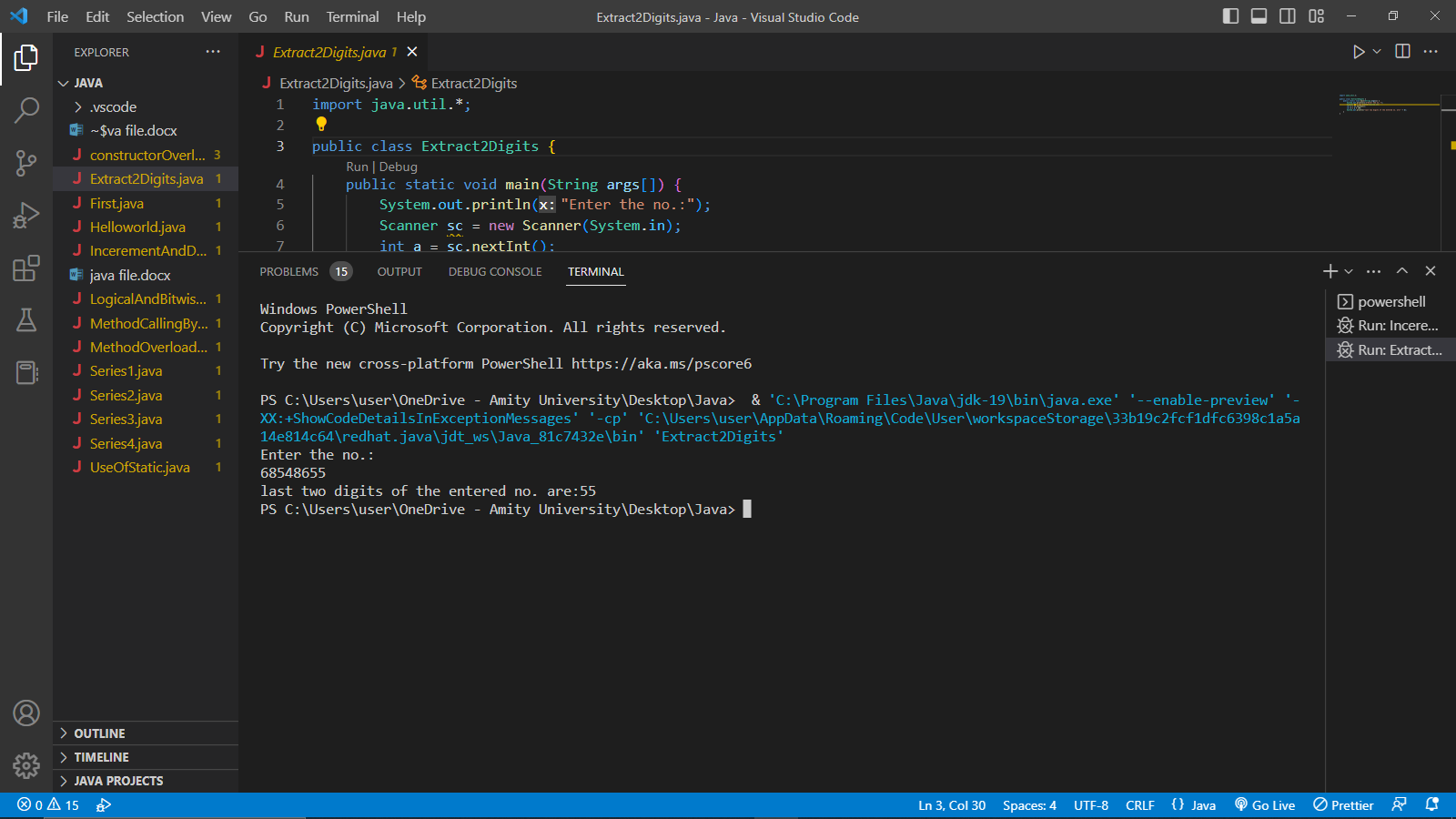
int b = a % 100;

System.out.println("last two digits of the entered no. are:" + b);

}

}

**Output:**



PROGRAM – 05

**Problem Statement:**

Write a program in java to find the sum of the given series up to certain number of terms.

Series: 1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9 +…

**Source Code:**

import java.util.\*;

class Series1 {

public static void main(String args[]) {

System.out.println("Enter the no. of terms:");

Scanner sc = new Scanner(System.in);

int a = sc.nextInt();

int k = 0;

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

k += i;

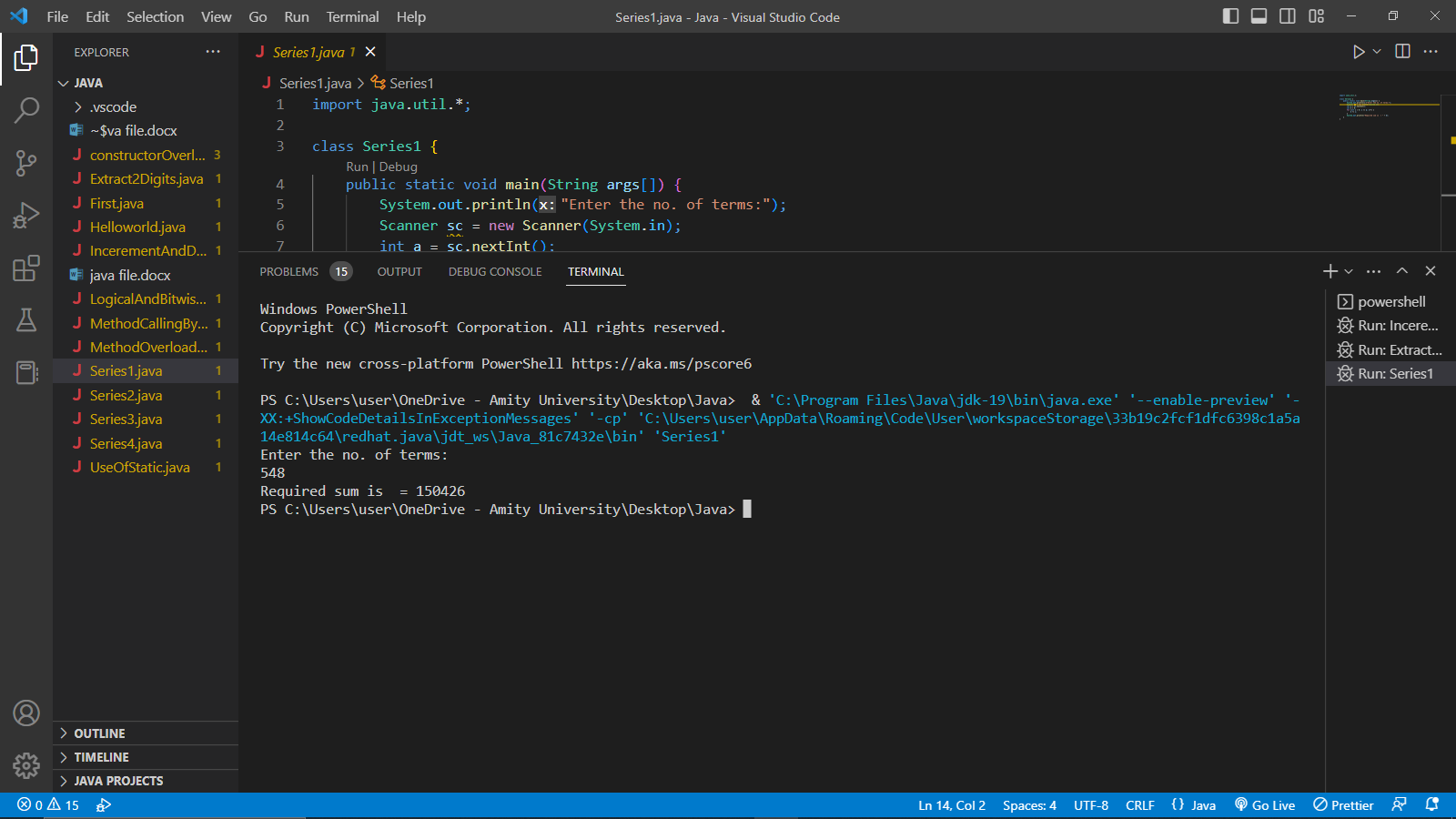
}

System.out.println("Required sum is = " + k);

}

}

**Output:**



PROGRAM – 06

**Problem Statement:**

Write a program in java to find the sum of the given series up to certain number of terms.

Series: 5 + 6 + 7/8 + 1/2 + 2/3 + 3/4 + 4/5 +…

**Source Code:**

import java.util.\*;

class Series2 {

public static void main(String args[]) {

System.out.println("Enter the no. of terms:");

Scanner sc = new Scanner(System.in);

int a = sc.nextInt();

double k = 5 + 6 + 7 / 8;

for (double i = 3; i <= a; i++) {

k += ((i - 3) / (i - 2));

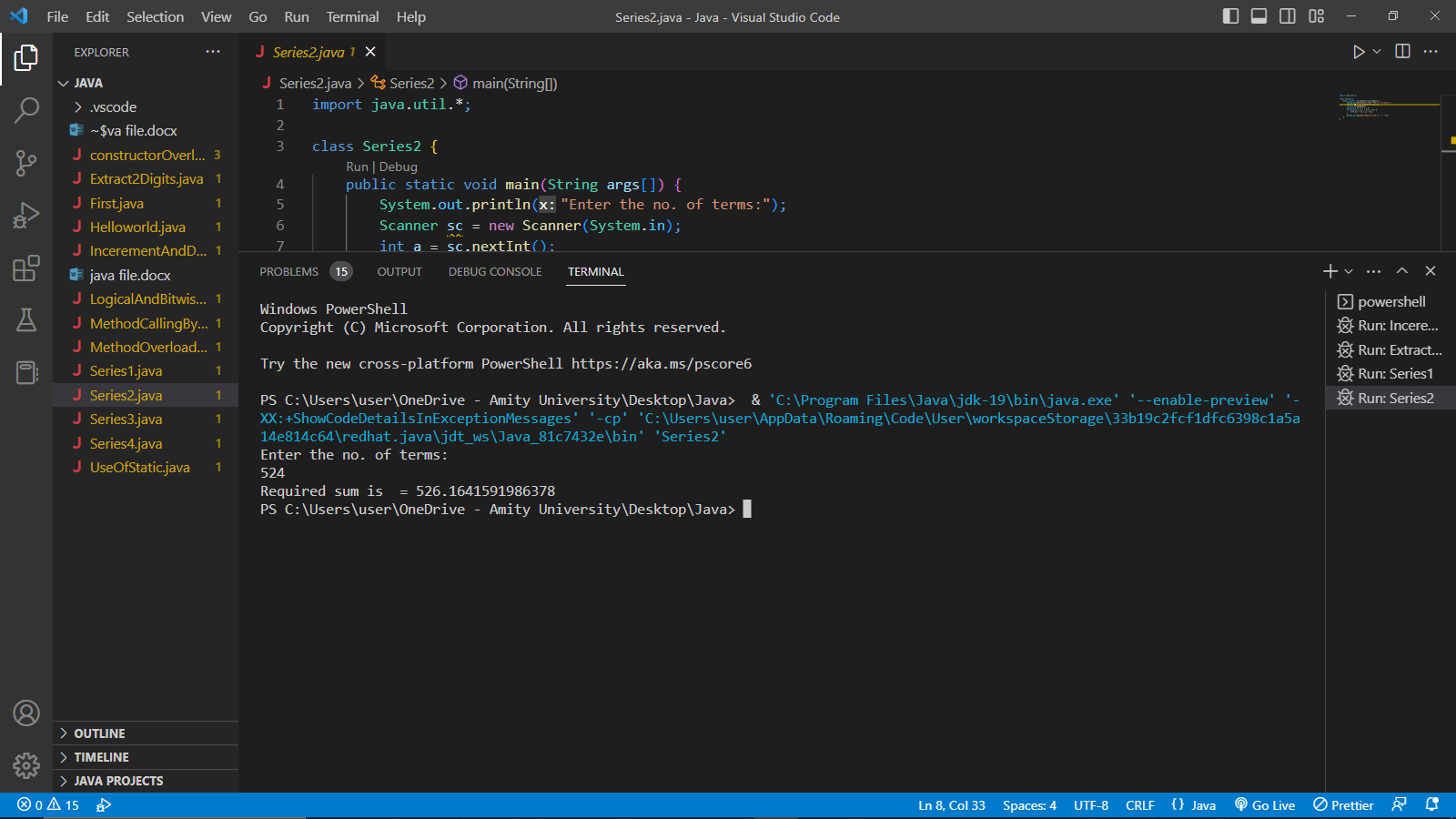
}

System.out.println("Required sum is = " + k);

}

}

**Output:**



PROGRAM – 07

**Problem Statement:**

Write a program in java to find the sum of the given series up to certain number of terms.

Series: 1 - 1/2 + 1/4 - 1/6 + 1/8 - 1/10 +…

**Source Code:**

import java.util.\*;

class Series3 {

public static void main(String args[]) {

System.out.println("Enter the no. of terms:");

Scanner sc = new Scanner(System.in);

int a = sc.nextInt();

double k = 1;

for (double i = 2; i <= a; i++) {

k += (Math.pow(-1, i + 1) / 2 \* (i - 1));

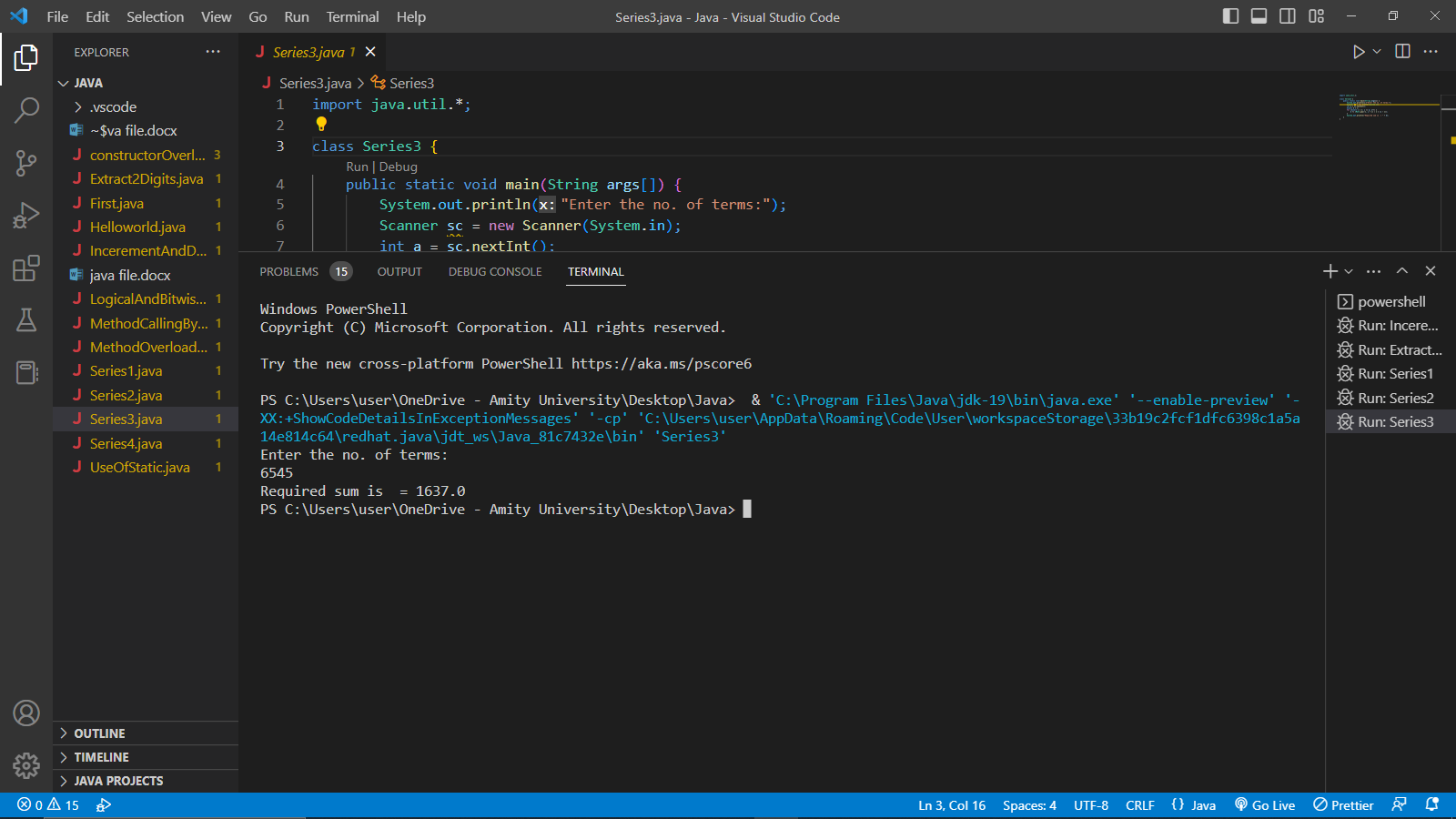
}

System.out.println("Required sum is = " + k);

}

}

**Output:**



PROGRAM – 08

**Problem Statement:**

Write a program in java to find the sum of the given series up to certain number of terms.

Series: 1 + 1/2! + 1/4! + 1/6! + 1/8! + 1/10! +…

**Source Code:**

import java.util.\*;

class Series4 {

public static void main(String args[])

{

System.out.println("Enter the no. of terms:");

Scanner sc = new Scanner(System.in);

int a = sc.nextInt();

double k = 0;

Series4 obj = new Series4();

for (double i = 2; i <= a; i++) {

k += (1 / obj.factorial(2 \* (i - 1)));

}

System.out.println("Required sum is = " + k);

}

double factorial(double n)

{

double l = 1;

for (double j = 1; j <= n; j++) {

l = l \* j;

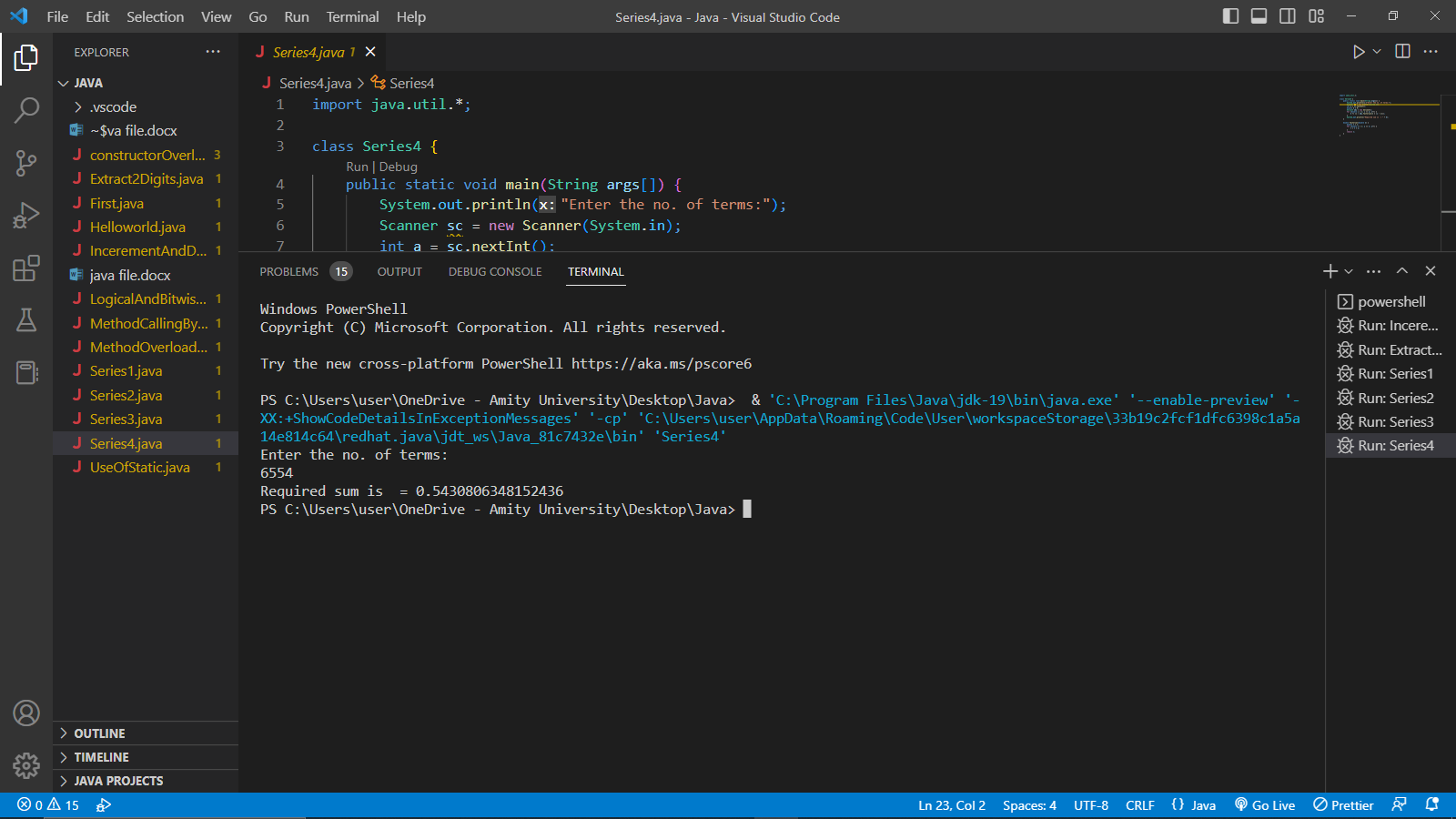
}

return l;

}

}

**Output:**



PROGRAM – 09

**Problem Statement:**

Write a program in java to find the sum of the given series up to certain number of terms. [showing method calling within two classes]

Series: 1 + 1/2! + 1/4! + 1/6! + 1/8! + 1/10! +…

**Source Code:**

import java.util.\*;

class MethodCallingByDifferentClass {

public static void main(String args[]) {

System.out.println("Enter the no. of terms:");

Scanner sc = new Scanner(System.in);

int a = sc.nextInt();

double k = 0;

Factorial obj = new Factorial();

for (double i = 2; i <= a; i++) {

k += (1 / obj.factorial(2 \* (i - 1)));

}

System.out.println("Required sum is = " + k);

}

}

class Factorial {

double factorial(double n) {

double l = 1;

for (double j = 1; j <= n; j++) {

l = l \* j;

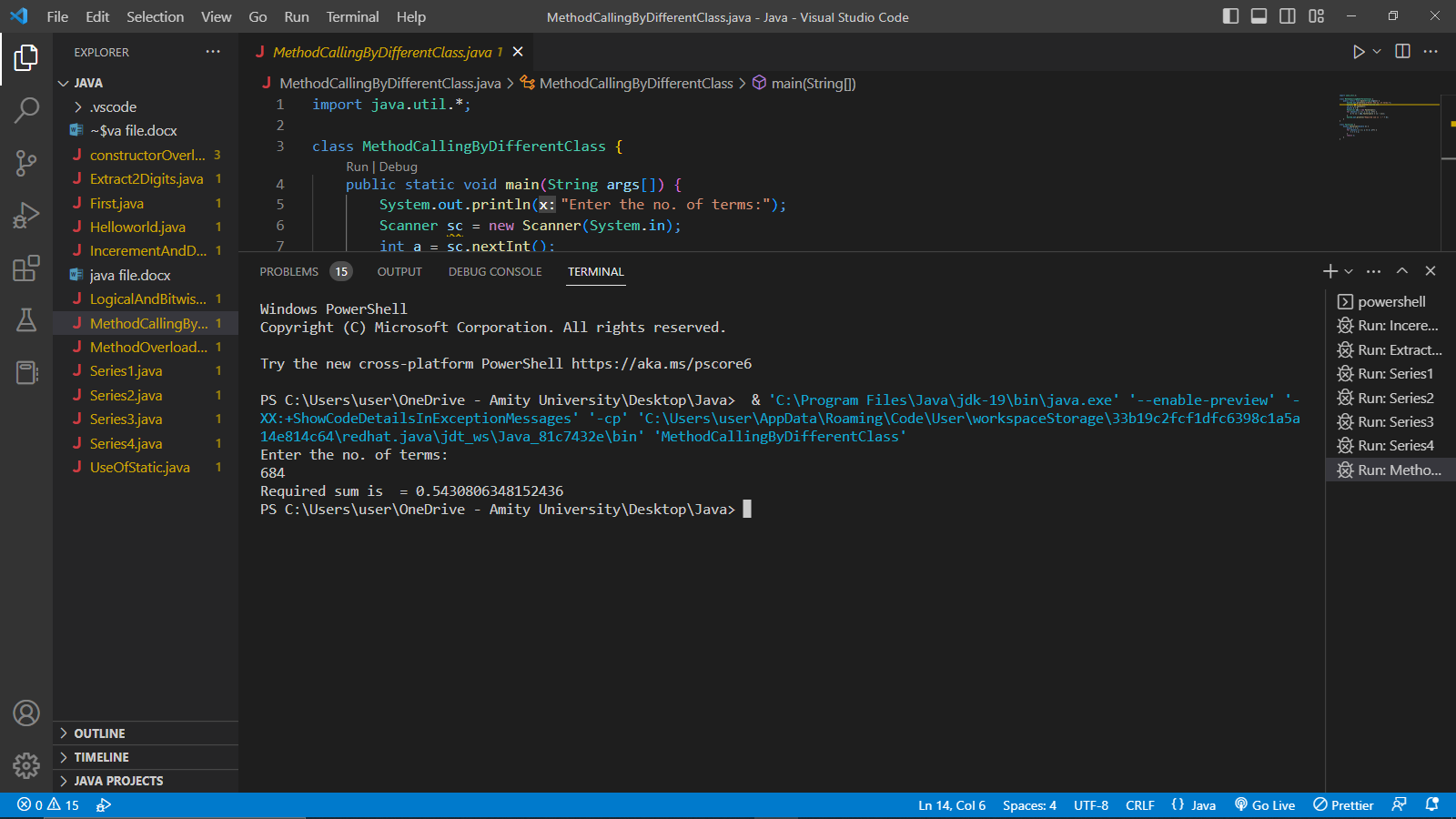
}

return l;

}

}

**Output:-**



PROGRAM – 10

**Problem Statement:**

Write a program in java to find the sum of the given series up to certain number of terms. [showing use of static in classes]

Series: 1 + 1/2! + 1/4! + 1/6! + 1/8! + 1/10! +…

**Source Code:**

import java.util.\*;

public class UseOfStatic {

class MethodCallingByDifferentClass {

public static void main(String args[]) {

System.out.println("Enter the no. of terms:");

Scanner sc = new Scanner(System.in);

int a = sc.nextInt();

double k = 0;

for (double i = 2; i <= a; i++) {

k += (1 / Factorial.factorial(2 \* (i - 1)));

}

System.out.println("Required sum is = " + k);

}

}

class Factorial {

static double factorial(double n) {

double l = 1;

for (double j = 1; j <= n; j++) {

l = l \* j;

}

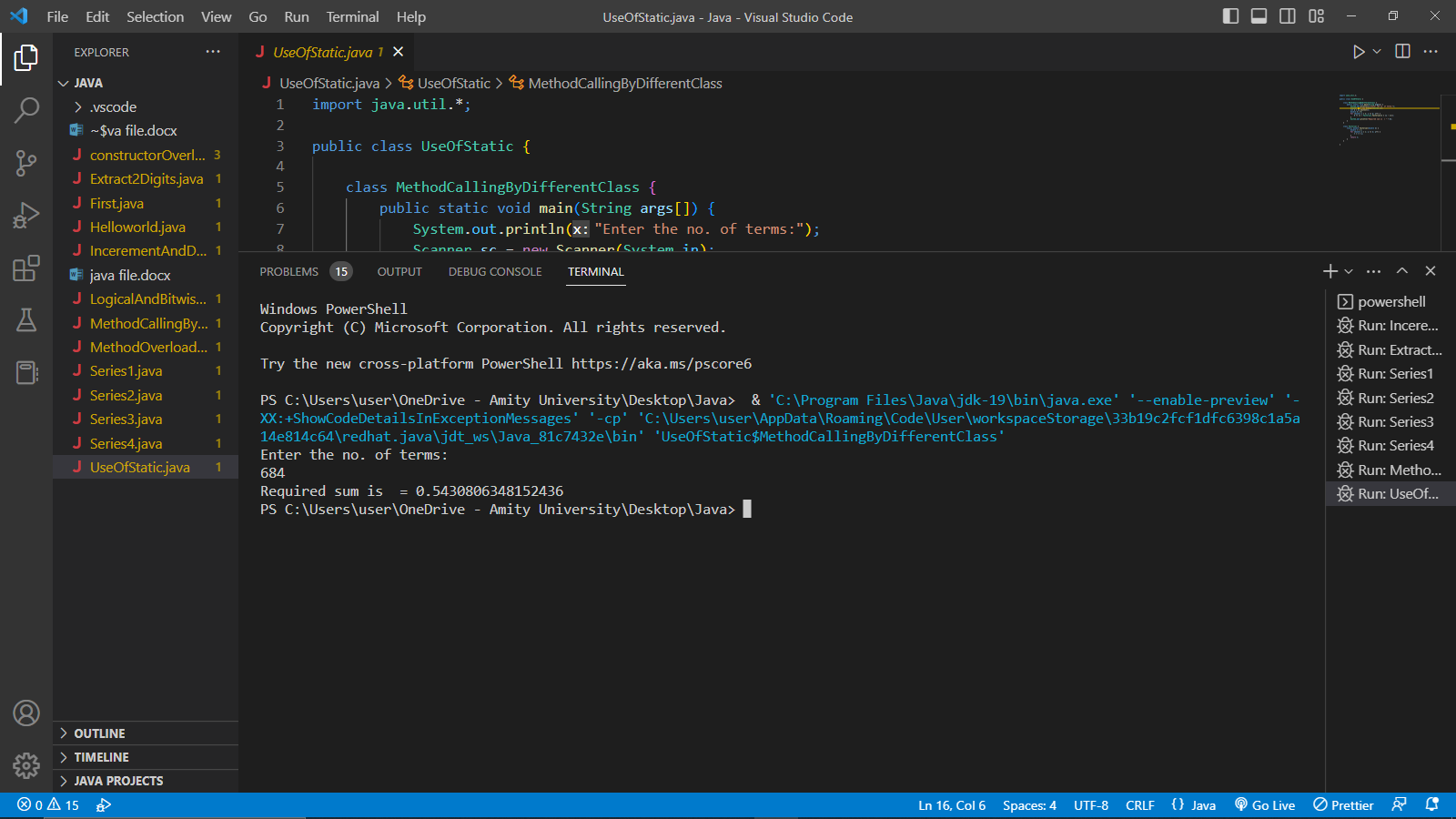
return l;

}

}

}

**Output:-**



PROGRAM – 11

**Problem Statement:**

Write a program in java to calculate the area of a rectangle and a circle [using method overloading].

**Source Code:**

import java.util.\*;

public class MethodOverloading {

public static void main(String args[]) {

System.out.print("Enter the length and breadth of rectangle:");

Scanner sc = new Scanner(System.in);

int l = sc.nextInt();

int b = sc.nextInt();

int ar = Area.area(l, b);

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

System.out.println("Enter the radius of circle:");

double r = sc.nextDouble();

double arc = Area.area(r);

System.out.print("Area of circle is:" + arc);

}

}

class Area {

static int area(int l, int b) {

int ar = l \* b;

return ar;

}

static double area(double r) {

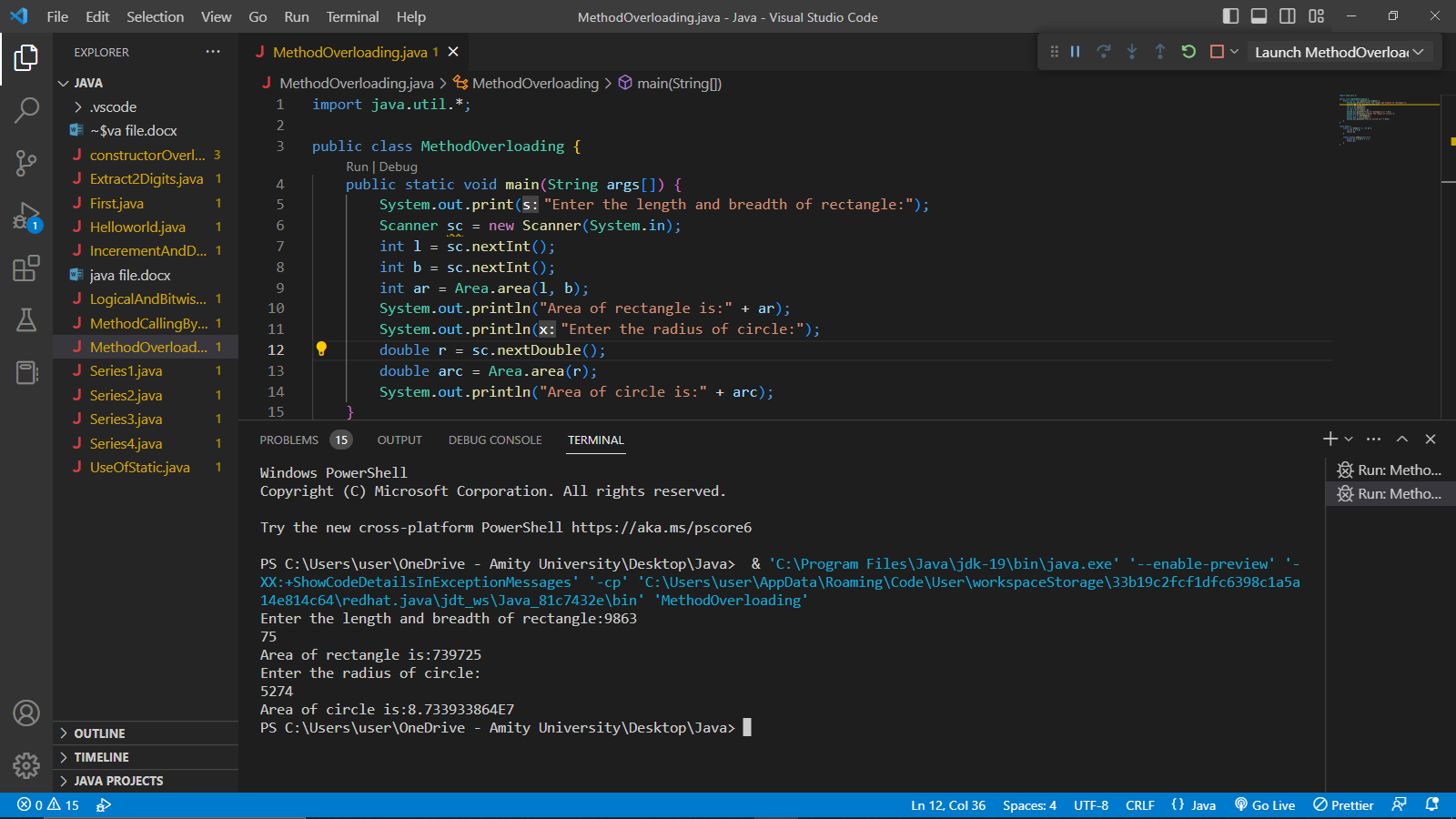
double ar = 3.14 \* r \* r;

return ar;

}

}

**Output:-**



PROGRAM – 12

**Problem Statement:**

Write a program in java to calculate the area of a rectangle and a circle [using constructor overloading].

**Source Code:**

import java.util.\*;

public class constructorOverloadingArea

{

public static void main(String args[])

{

System.out.print("Enter the length and breadth of rectangle:");

Scanner sc = new Scanner(System.in);

int l = sc.nextInt();

int b = sc.nextInt();

constructorOverloadingArea objR = new constructorOverloadingArea(l, b);

System.out.println("Enter the radius of circle:");

double r = sc.nextDouble();

constructorOverloadingArea objC = new constructorOverloadingArea(r);

}

constructorOverloadingArea(int l, int b) {

int ar = l \* b;

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

}

constructorOverloadingArea(double r) {

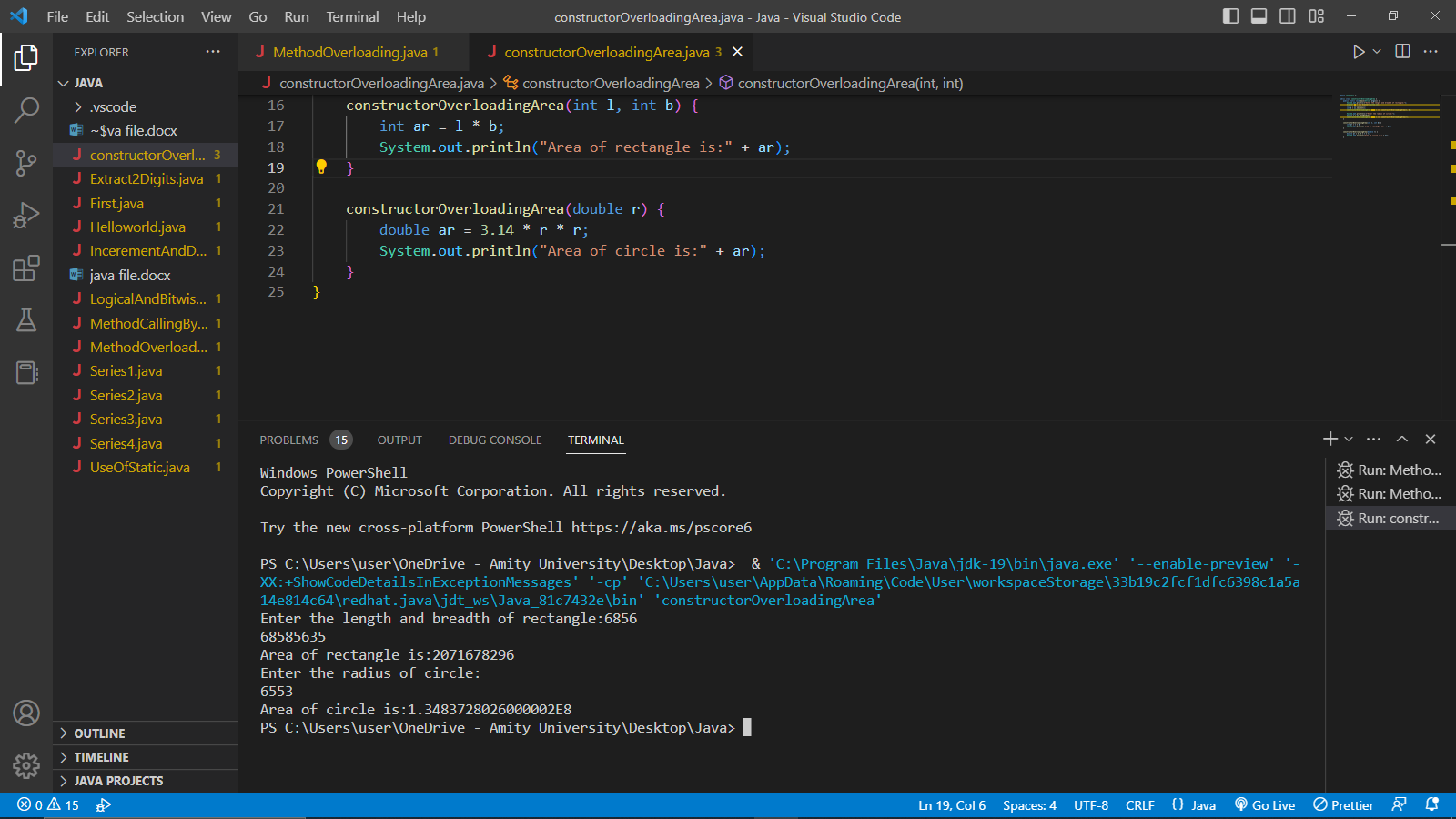
double ar = 3.14 \* r \* r;

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

}

}

**Output:-**



PROGRAM – 13

**Problem Statement:**

Write a program in java to check whether the given no is Armstrong Number.

**Source Code:**

import java.util.\*;

class ArmstrongNumber {

public static void main(String args[]) {

System.out.println("Enter the no.: ");

Scanner sc = new Scanner(System.in);

int a = sc.nextInt();

int temp = a;

int sum = 0;

while (a > 0) {

int d = a % 10;

a /= 10;

sum += Math.pow(d, 3);

}

if (sum == temp) {

System.out.println("Given number is an Armstorng number.");

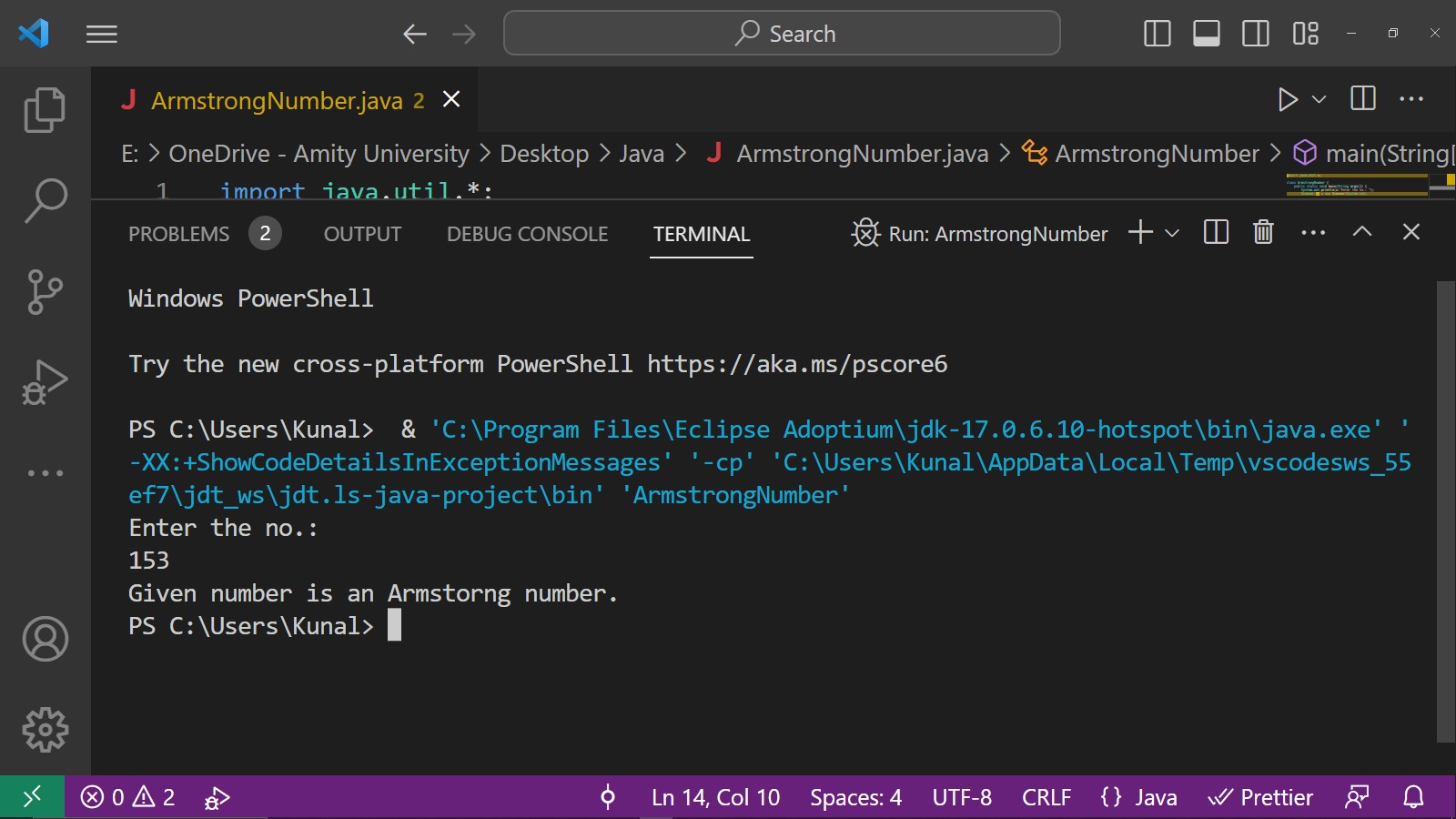
} else {

System.out.println("Given number is not an Armstorng number.");

}

}

}

**Output:** 

PROGRAM – 14

**Problem Statement:**

Write a program in java to calculate the sum of square of the digits placed at the even positions of the umber entered by the user.

**Source Code:**

import java.util.\*;

public class sumOfSquareOfEvenPosition {

public static void main(String args[]) {

System.out.println("Enter the no.:");

Scanner sc = new Scanner(System.in);

int a = sc.nextInt();

int sum = 0;

int i = 0;

int d = 0;

while (a > 0) {

d = a % 10;

a = a / 10;

}

if (i % 2 == 0) {

sum = sum + d \* d;

}

System.out.println("Required sum is : " + sum);

}

}

**Output:** A screenshot of a computer

Description automatically generated

PROGRAM – 15

**Problem Statement:**

Write a program in java to check whether the given input is Palindrome.

**Source Code:**

import java.util.\*;

public class Palindrome {

public static void main(String args[]) {

System.out.println("Enter the no.:");

Scanner sc = new Scanner(System.in);

int a = sc.nextInt();

int temp = a;

int d = 0;

int rev = 0;

while (a > 0) {

d = a % 10;

rev = rev \* 10 + d;

a = a / 10;

}

if (temp == rev) {

System.out.print("Entered number is a Palindrome:");

} else {

System.out.print("Entered number is not a Palindrome:");

}

}

}

**Output:**A screenshot of a computer

Description automatically generated

PROGRAM – 16

**Problem Statement:**

Write a program in java with a class named Box with the data members as length, breadth and height, and member functions as area() and volume(), also create another class containing main method to create objects and call the methods of the Box class.

**Source Code:**

class Box {

double height;

double length;

double weight;

public double area() {

return 2 \* (length \* height + height \* weight + length \* weight);

}

public double volume() {

return length \* height \* weight;

}

}

public class BoxDemo {

public static void main(String[] args) {

Box box1 = new Box();

box1.height = 10;

box1.length = 20;

box1.weight = 30;

double area = box1.area();

double volume = box1.volume();

System.out.println("Area: " + area);

System.out.println("Volume: " + volume);

} }

**Output:**

A screenshot of a computer

Description automatically generated

PROGRAM – 17

**Problem Statement:**

Write a program in java to take input of 10 elements of an array and print the array also.

**Source Code:**

import java.util.Scanner;

public class Array {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

int[] arr = new int[10];

// Input 10 elements of the array

for (int i = 0; i < 10; i++) {

System.out.print("Enter element " + (i+1) + ": ");

arr[i] = scanner.nextInt();

}

// Print the array

System.out.print("The array is: [ ");

for (int i = 0; i < 10; i++) {

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

}

System.out.println("]");

}

}

**Output:** A screenshot of a computer

Description automatically generated

PROGRAM – 18

**Problem Statement:**

Write a program in java to calculate the sum of elements arrays of the elements.

**Source Code:**

import java.util.Scanner;

public class ArraySum {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

int[] arr = new int[10];

int sum = 0;

// Input 10 elements of the array

for (int i = 0; i < 10; i++) {

System.out.print("Enter element " + (i+1) + ": ");

arr[i] = scanner.nextInt();

}

// Calculate the sum of the elements in the array

for (int i = 0; i < 10; i++) {

sum += arr[i];

}

// Print the sum of the elements in the array

System.out.println("The sum of the elements in the array is: " + sum);

}

}

**Output:**

A screenshot of a computer

Description automatically generated

PROGRAM – 19

**Problem Statement:**

Write a program in java for the following searching algorithms:

1. Linear Search

**Source Code:**

public class LinearSearch {

public static int linearSearch(int[] arr, int target) {

for (int i = 0; i < arr.length; i++) {

if (arr[i] == target) {

return i;

}

}

return -1;

}

public static void main(String[] args) {

int[] arr = {1, 2, 3, 4, 5};

int target = 3;

int index = linearSearch(arr, target);

if (index == -1) {

System.out.println("Target not found");

} else {

System.out.println("Target found at index " + index);

}

}

}

**Output:** A screenshot of a computer

Description automatically generated

1. Binary Search

**Source Code:**

public class BinarySearch {

public static int binarySearch(int[] arr, int target) {

int left = 0;

int right = arr.length - 1;

while (left <= right) {

int mid = (left + right) / 2;

if (arr[mid] == target) {

return mid;

} else if (arr[mid] < target) {

left = mid + 1;

} else {

right = mid - 1;

}

}

return -1;

}

public static void main(String[] args) {

int[] arr = {1, 2, 3, 4, 5};

int target = 3;

int index = binarySearch(arr, target);

if (index == -1) {

System.out.println("Target not found");

} else {

System.out.println("Target found at index " + index);

}

}

}

**Output:**

A screenshot of a computer

Description automatically generated

PROGRAM – 20

**Problem Statement:**

Write a program in java for the following sorting algorithms:

1. Bubble Sort

**Source Code:**

public class BubbleSort {

public static void bubbleSort(int[] arr) {

int n = arr.length;

for (int i = 0; i < n - 1; i++) {

for (int j = 0; j < n - i - 1; j++) {

if (arr[j] > arr[j + 1]) {

int temp = arr[j];

arr[j] = arr[j + 1];

arr[j + 1] = temp;

}

}

}

}

public static void main(String[] args) {

int[] arr = {5, 4, 3, 2, 1};

bubbleSort(arr);

for (int i = 0; i < arr.length; i++) {

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

}

}

}

**Output:** A screenshot of a computer

Description automatically generated

0

1. Insertion Sort

**Source Code:**

public class InsertionSort {

public static void insertionSort(int[] arr) {

int n = arr.length;

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

int key = arr[i];

int j = i - 1;

while (j >= 0 && arr[j] > key) {

arr[j + 1] = arr[j];

j--;

}

arr[j + 1] = key;

}

}

public static void main(String[] args) {

int[] arr = {5, 4, 3, 2, 1};

insertionSort(arr);

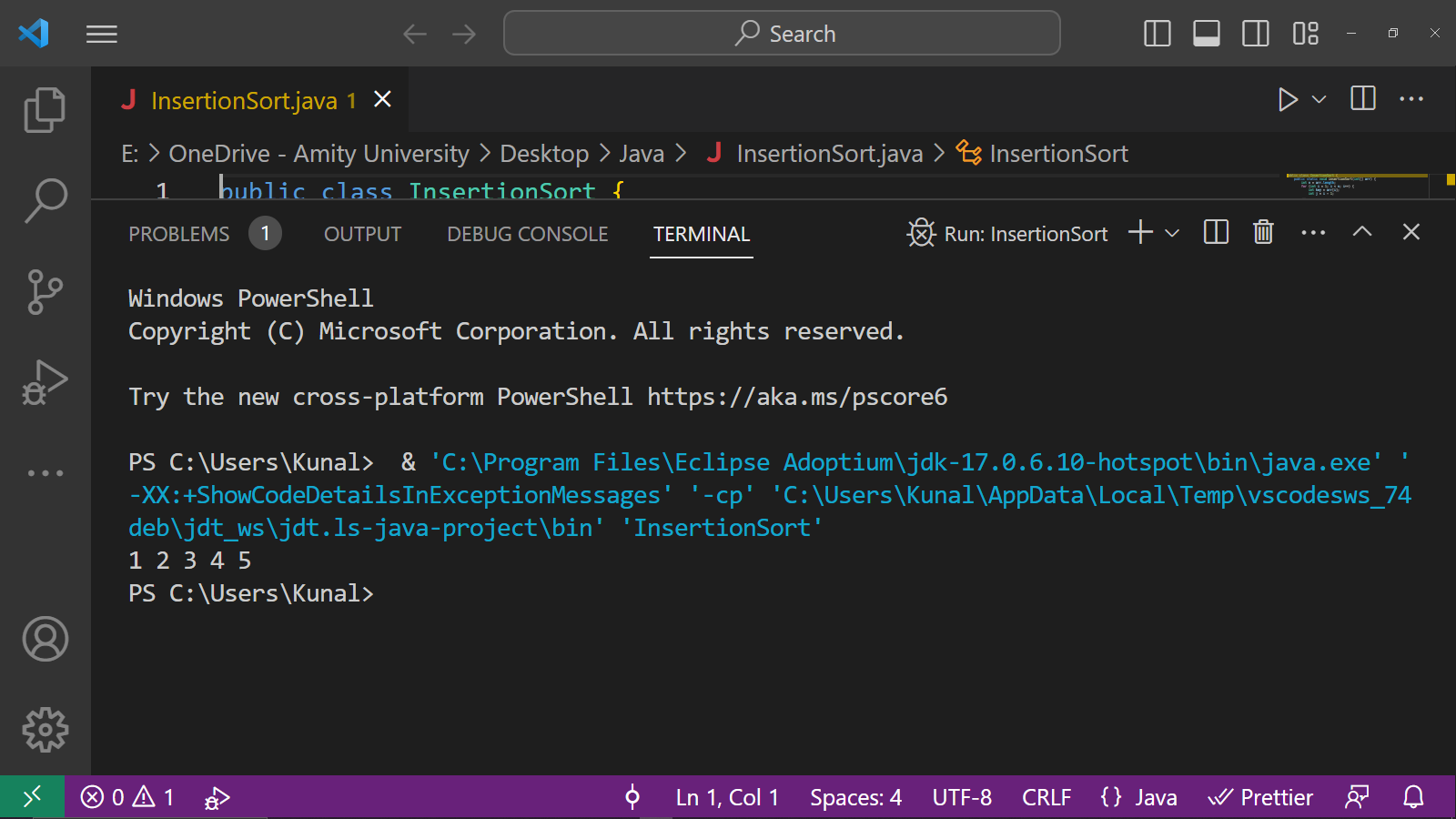
for (int i = 0; i < arr.length; i++) {

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

}

}

}

**Output:** 

PROGRAM – 21

**Problem Statement:**

Write a program in java for the various types of inheritance.

**Source Code:**

// Base class

class Animal {

public void eat() {

System.out.println("The animal is eating.");

}

}

// Single inheritance

class Dog extends Animal {

public void bark() {

System.out.println("The dog is barking.");

}

}

// Multilevel inheritance

class BabyDog extends Dog {

public void weep() {

System.out.println("The baby dog is weeping.");

}

}

// Hierarchical inheritance

class Cat extends Animal {

public void meow() {

System.out.println("The cat is meowing.");

}

}

// Multiple inheritance (achieved using interfaces)

interface Flyable {

void fly();

}

interface Swimmable {

void swim();

}

class Duck implements Flyable, Swimmable {

public void fly() {

System.out.println("The duck is flying.");

}

public void swim() {

System.out.println("The duck is swimming.");

}

}

public class InheritanceDemo {

public static void main(String[] args) {

// Single inheritance

Dog dog = new Dog();

dog.eat();

dog.bark();

// Multilevel inheritance

BabyDog babyDog = new BabyDog();

babyDog.eat();

babyDog.bark();

babyDog.weep();

// Hierarchical inheritance

Cat cat = new Cat();

cat.eat();

cat.meow();

// Multiple inheritance

Duck duck = new Duck();

duck.fly();

duck.swim();

}

}

**Output:** A screenshot of a computer

Description automatically generated with medium confidence

PROGRAM – 22

**Problem Statement:**

Write a Program in java to show the implementation of threads

**Source Code:**

public class RunExp3 extends Thread{

public void run(){

for(int i=1;i<6;i++)

{ try

{

Thread.sleep(500);

}catch(InterruptedException e){System.out.println(e);}

System.out.println(i);

}

}

public static void main(String args[]){

RunExp3 t1=new RunExp3();

RunExp3 t2=new RunExp3();

t1.run();

t2.run();

}

}

**Output:** 2

1

2

3

4

5

1

2

3

4

5

3

4PROGRAM – 23

**Problem Statement:**

Write a program in java to show the implementation of try-catch-finally keywords.

**Source Code:**

import java.io.\*;

class ListOfNumbers {

// create an integer array

private int[] list = {5, 6, 8, 9, 2};

// method to write data from array to a fila

public void writeList() {

PrintWriter out = null;

try {

System.out.println("Entering try statement");

// creating a new file OutputFile.txt

out = new PrintWriter(new FileWriter("OutputFile.txt"));

// writing values from list array to Output.txt

for (int i = 0; i < 7; i++) {

out.println("Value at: " + i + " = " + list[i]);

}

}

catch (Exception e) {

System.out.println("Exception => " + e.getMessage());

}

finally {

// checking if PrintWriter has been opened

if (out != null) {

System.out.println("Closing PrintWriter");

// close PrintWriter

out.close();

}

else {

System.out.println("PrintWriter not open");

}

}

}

}

class Main {

public static void main(String[] args) {

ListOfNumbers list = new ListOfNumbers();

list.writeList();

}

}

**Output:**

Entering try statement

Exception => Index 5 out of bounds for length 5

Closing PrintWriter

4

PROGRAM – 24

**Problem Statement:**

Write a program in java to show the implementation of throw and throws.

**Source Code:**

**Throw**

import java.io.\*;

class Main {

public static void findFile() throws IOException {

// code that may produce IOException

File newFile=new File("test.txt");

FileInputStream stream=new FileInputStream(newFile);

}

public static void main(String[] args) {

try{

findFile();

} catch(IOException e){

System.out.println(e);

}

}

}

**Output:**

java.io.FileNotFoundException: test.txt (No such file or directory)

**Throw s**

class Main {

public static void divideByZero() {

throw new ArithmeticException("Trying to divide by 0");

}

public static void main(String[] args) {

divideByZero();

}

}

**Output:**

Exception in thread "main" java.lang.ArithmeticException: Trying to divide by 0

at Main.divideByZero(Main.java:3)

at Main.main(Main.java:7)

exit status 1

4PROGRAM – 25

**Problem Statement:**

Write a program in java AWT to design a login page

**Source Code:**

import java.awt.\*;

import java.awt.event.\*;

class MyLoginWindow extends Frame{

TextField name,pass;

Button b1,b2;

MyLoginWindow()

{

setLayout(new FlowLayout());

this.setLayout(null);

Label n=new Label("Name:",Label.CENTER);

Label p=new Label("password:",Label.CENTER);

name=new TextField(20);

pass=new TextField(20);

pass.setEchoChar('#');

b1=new Button("submit");

b2=new Button("cancel");

this.add(n);

this.add(name);

this.add(p);

this.add(pass);

this.add(b1);

this.add(b2);

n.setBounds(70,90,90,60);

p.setBounds(70,130,90,60);

name.setBounds(200,100,90,20);

pass.setBounds(200,140,90,20);

b1.setBounds(100,260,70,40);

b2.setBounds(180,260,70,40);

}

public static void main(String args[])

{

MyLoginWindow ml=new MyLoginWindow();

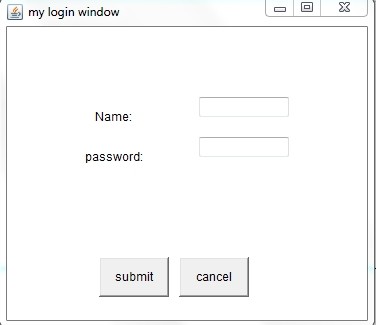
ml.setVisible(true);

ml.setSize(400,400);

ml.setTitle("my login window");

}

}

**Output:**