Assignments Module – 6 (Core Java)

1. W.A.J.P to Take three numbers from the user and print the greatest number.

Code:

import java.util.Scanner;

public class GreatestNumber {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.println("Enter the first number:");

int num1 = scanner.nextInt();

System.out.println("Enter the second number:");

int num2 = scanner.nextInt();

System.out.println("Enter the third number:");

int num3 = scanner.nextInt();

int greatestNumber = num1;

if (num2 > greatestNumber) {

greatestNumber = num2;

}

if (num3 > greatestNumber) {

greatestNumber = num3;

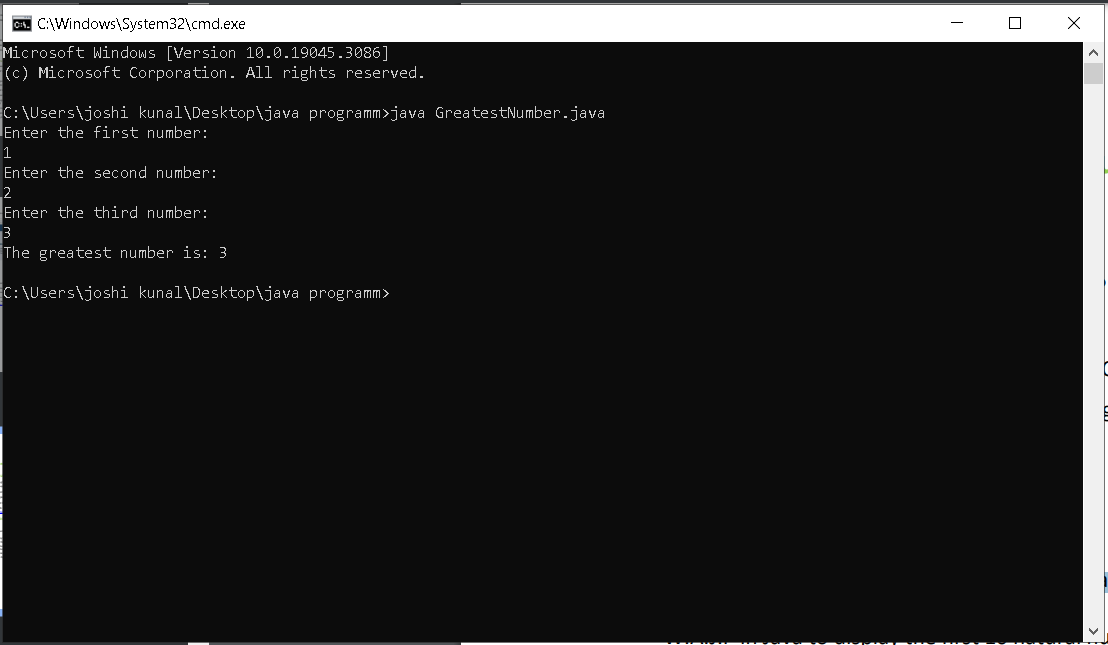
}

System.out.println("The greatest number is: " + greatestNumber);

}

}

Output:



1. W.A.J.P in Java to display the first 10 natural numbers using while loop.

Code:

public class FirstTenNaturalNumbers {

public static void main(String[] args) {

int i = 1;

while (i <= 10) {

System.out.println(i);

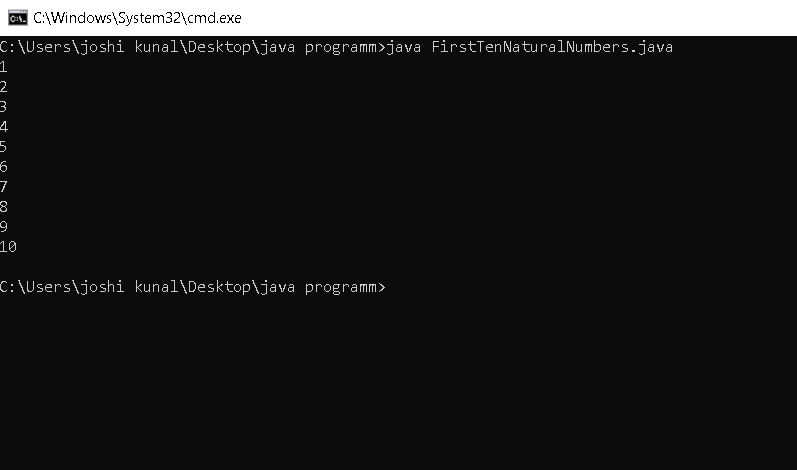
i++;

}

}

}

Output:



1. W.A.J.P to find factorial for Given Number.

Code:

import java.util.Scanner;

public class Factorial {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

int number = scanner.nextInt();

int factorial = 1;

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

factorial \*= i;

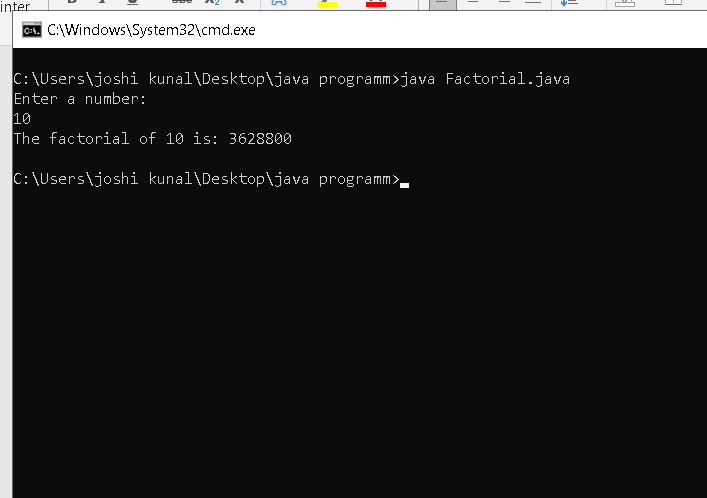
}

System.out.println("The factorial of " + number + " is: " + factorial);

}

}

Output:



1. W.A.J.P to check given number is Prime or not?

Code:

public class PrimeNumber {

public static void main(String[] args) {

int number = 11;

boolean isPrime = true;

for (int i = 2; i <= number / 2; i++) {

if (number % i == 0) {

isPrime = false;

break;

}

}

if (isPrime) {

System.out.println(number + " is a prime number");

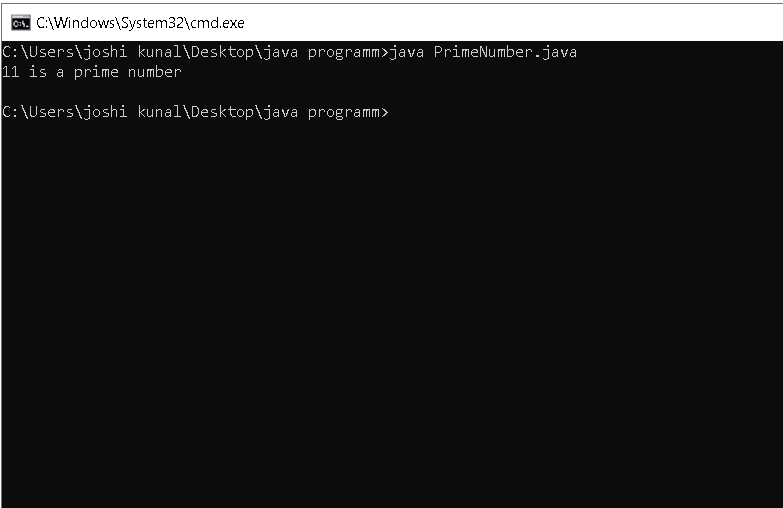
} else {

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

}

}

}

Output

1. W.A.J.P to check given number is Armstrong or not?

Code:

import java.util.Scanner;

public class ArmstrongNumber {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

int number = scanner.nextInt();

int originalNumber = number;

int sum = 0;

int temp = number;

while (temp > 0) {

int digit = temp % 10;

sum += Math.pow(digit, number);

temp = temp / 10;

}

if (sum == originalNumber) {

System.out.println(number + " is an Armstrong number");

} else {

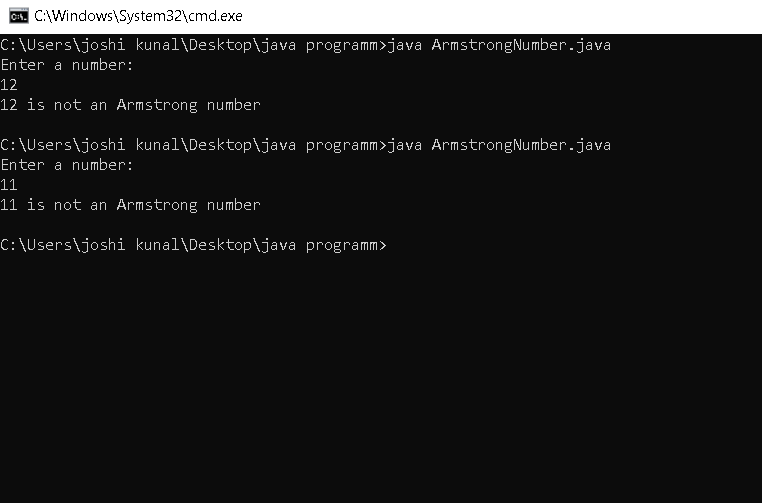
System.out.println(number + " is not an Armstrong number");

}

}

}

Output:



1. W.A.J.P for create Fibonacci Series.

Code:

public class FibonacciSeries {

public static void main(String[] args) {

int n = 10;

int firstTerm = 0;

int secondTerm = 1;

System.out.println("The first 10 Fibonacci numbers are:");

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

int nextTerm = firstTerm + secondTerm;

System.out.println(firstTerm);

firstTerm = secondTerm;

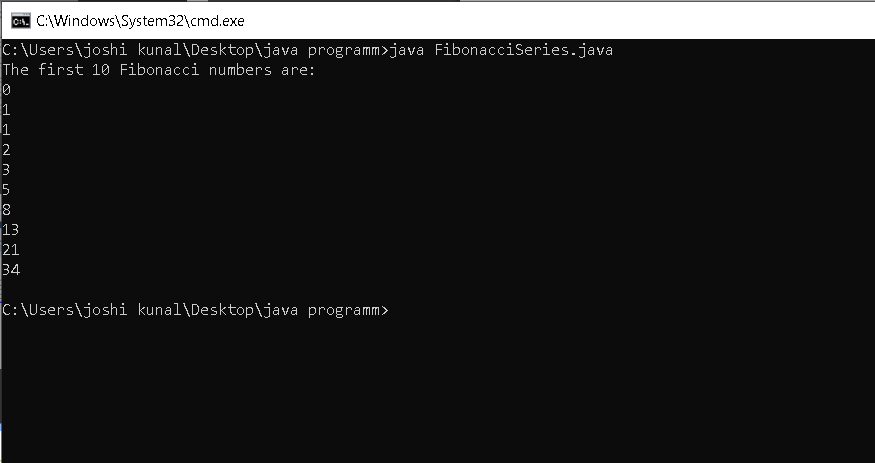
secondTerm = nextTerm;

}

}

}

Output:



1. W.A.J.P to Print pattern Given Below.

Code:

import java.util.Scanner;

public class Pattern {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

int rows = scanner.nextInt();

// Pattern 1

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

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

System.out.print(j);

}

System.out.println();

}

// Pattern 2

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

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

System.out.print(i);

}

System.out.println();

}

// Pattern 3

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

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

if (j % 2 == 0) {

System.out.print("1");

} else {

System.out.print("0");

}

}

System.out.println();

}

// Pattern 4

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

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

System.out.print(j);

}

for (int j = i - 1; j >= 1; j--) {

System.out.print(j);

}

System.out.println();

}

// Pattern 5

for (int i = rows; i >= 1; i--) {

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

System.out.print("\*");

}

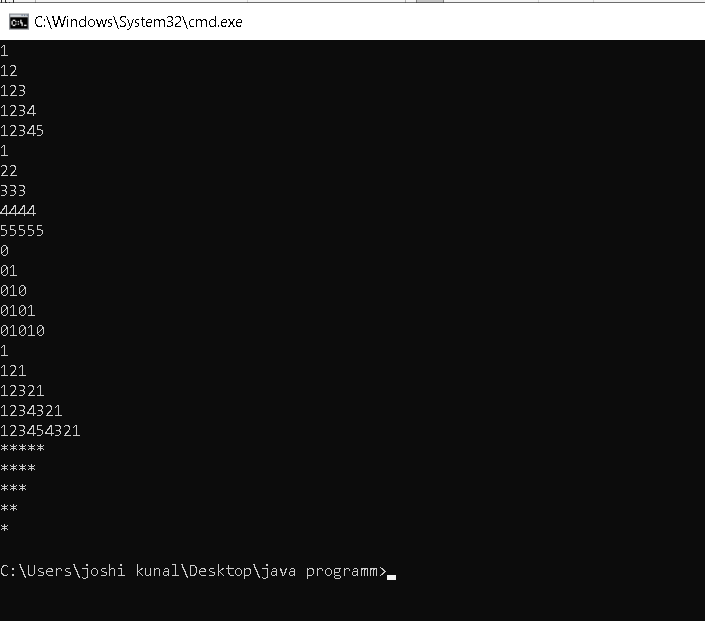
System.out.println();

}

}

}

Output:



1. WAP to compute the sum of the first 100 prime numbers.

Code:

import java.util.ArrayList;

import java.util.Scanner;

public class PrimeNumberSum {

public static void main(String[] args) {

ArrayList<Integer> primeNumbers = new ArrayList<>();

int sum = 0;

Scanner scanner = new Scanner(System.in);

System.out.println("Enter the number of prime numbers:");

int numberOfPrimeNumbers = scanner.nextInt();

for (int i = 2; i <= numberOfPrimeNumbers; i++) {

boolean isPrime = true;

for (int j = 2; j < i; j++) {

if (i % j == 0) {

isPrime = false;

break;

}

}

if (isPrime) {

primeNumbers.add(i);

sum += i;

}

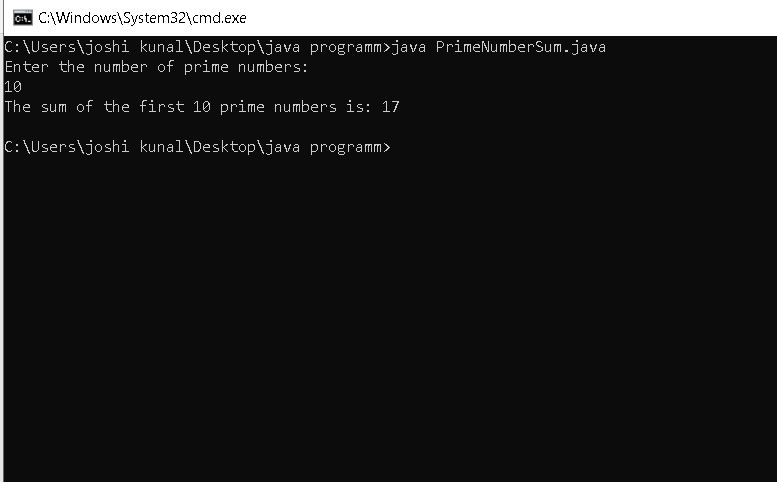
}

System.out.println("The sum of the first " + numberOfPrimeNumbers + " prime numbers is: " + sum);

}

}

Output:



1. WAP to sum values of an array.

Code:

public class SumOfArray {

public static void main(String[] args) {

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

int sum = 0;

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

sum += array[i];

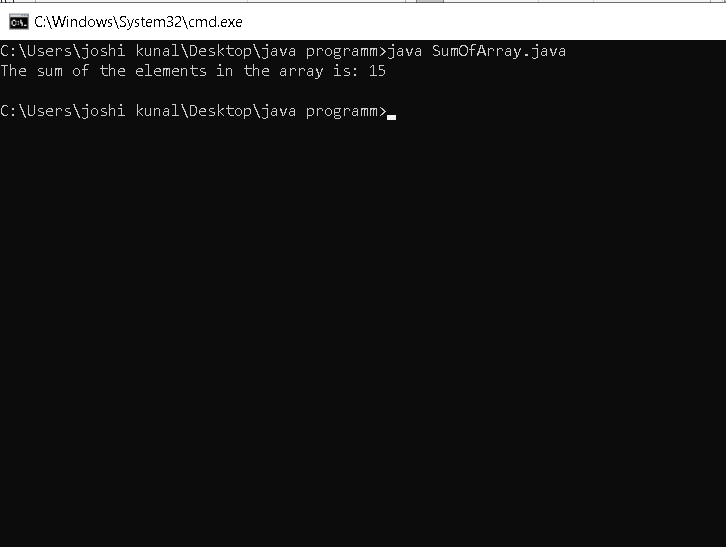
}

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

}

}

Output:



1. WAP to calculate the average value of array elements.

Code:

import java.util.Scanner;

public class AverageOfArray {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.println("Enter the number of elements in the array:");

int numberOfElements = scanner.nextInt();

int[] array = new int[numberOfElements];

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

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

array[i] = scanner.nextInt();

}

int sum = 0;

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

sum += array[i];

}

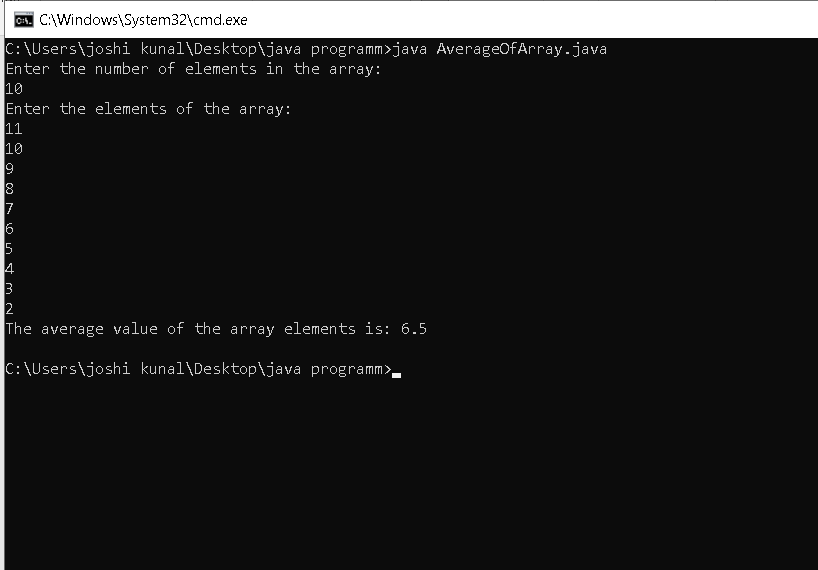
double average = (double) sum / numberOfElements;

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

}

}

Output:



1. WAP to find the index of an array element.

Code:

import java.util.Scanner;

public class ArrayIndex {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.println("Enter the number of elements in the array:");

int numberOfElements = scanner.nextInt();

int[] array = new int[numberOfElements];

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

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

array[i] = scanner.nextInt();

}

System.out.println("Enter the element you want to search for:");

int element = scanner.nextInt();

int index = -1;

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

if (array[i] == element) {

index = i;

break;

}

}

if (index == -1) {

System.out.println("The element " + element + " was not found in the array.");

} else {

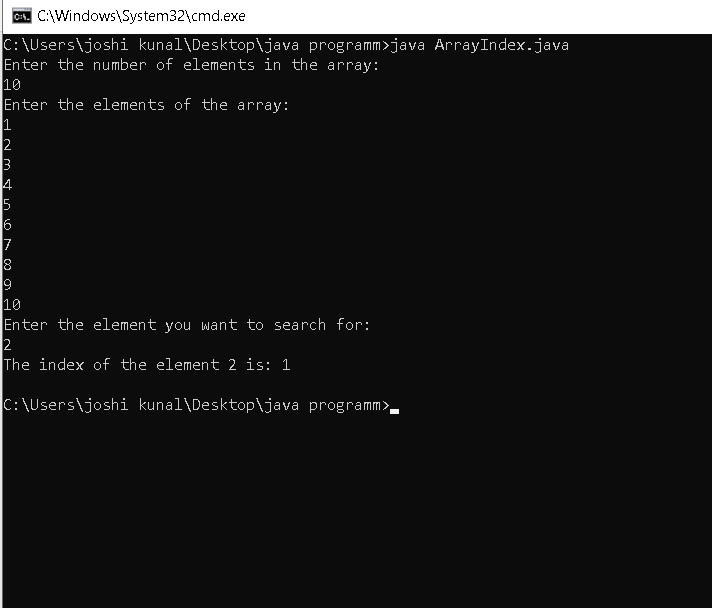
System.out.println("The index of the element " + element + " is: " + index);

}

}

}

Output:



1. WAP to find the maximum and minimum value of an array.

Code:

import java.util.Scanner;

public class ArrayMaximumMinimum {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.println("Enter the number of elements in the array:");

int numberOfElements = scanner.nextInt();

int[] array = new int[numberOfElements];

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

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

array[i] = scanner.nextInt();

}

int maxValue = array[0];

int minValue = array[0];

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

if (array[i] > maxValue) {

maxValue = array[i];

} else if (array[i] < minValue) {

minValue = array[i];

}

}

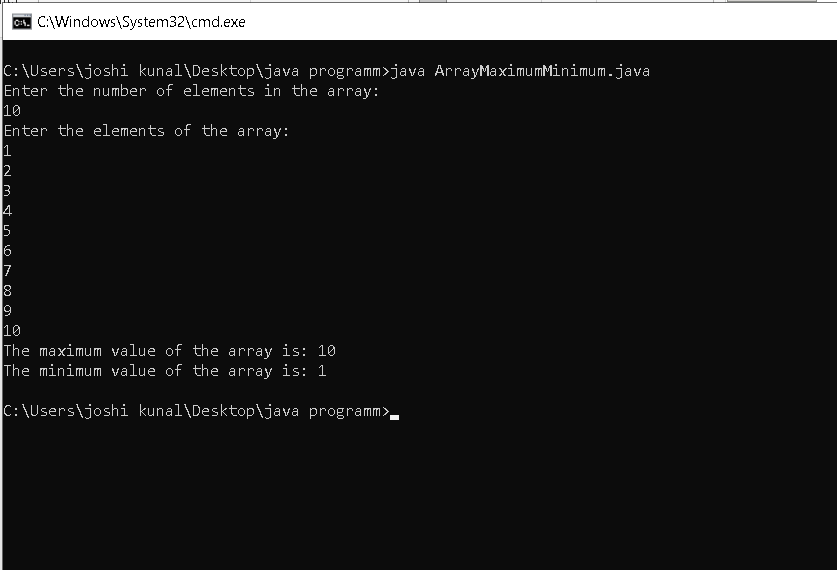
System.out.println("The maximum value of the array is: " + maxValue);

System.out.println("The minimum value of the array is: " + minValue);

}

}

Output:



1. WAP to Compare Two String

Code:

public class StringCompare {

public static void main(String[] args) {

String str1 = "Hello";

String str2 = "World";

if (str1.equals(str2)) {

System.out.println("The strings are equal.");

} else {

System.out.println("The strings are not equal.");

}

if (str1.compareTo(str2) < 0) {

System.out.println("The string " + str1 + " is less than the string " + str2);

} else if (str1.compareTo(str2) > 0) {

System.out.println("The string " + str1 + " is greater than the string " + str2);

} else {

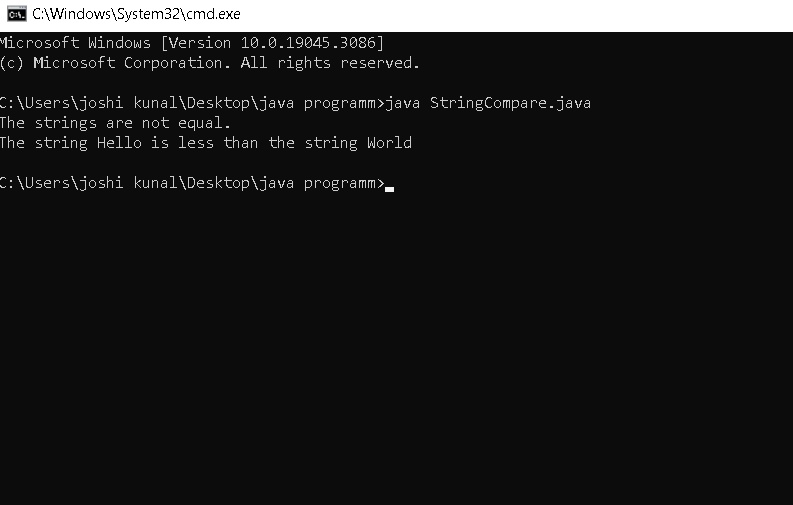
System.out.println("The strings are equal.");

}

}

}

Output:



1. WAP to concatenate a given string to the end of another string.

Code:

public class ConcatenateStrings {

public static void main(String[] args) {

String str1 = "Hello, ";

String str2 = "world!";

// Concatenate the two strings using the `concat()` method.

String str3 = str1.concat(str2);

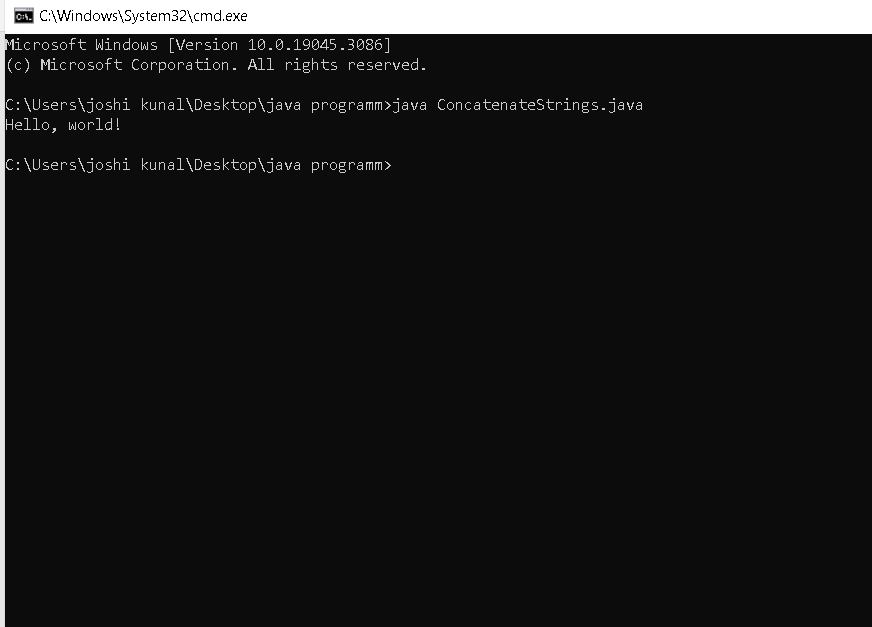
// Print the concatenated string.

System.out.println(str3);

}

}

Output:



1. WAP to demonstrate try catch block

Code:

public class TryCatchDemo {

public static void main(String[] args) {

int[] numbers = {1, 2, 3};

try {

// This code may throw an exception.

System.out.println(numbers[4]);

} catch (ArrayIndexOutOfBoundsException e) {

// This code will be executed if the exception is thrown.

System.out.println("The index is out of bounds.");

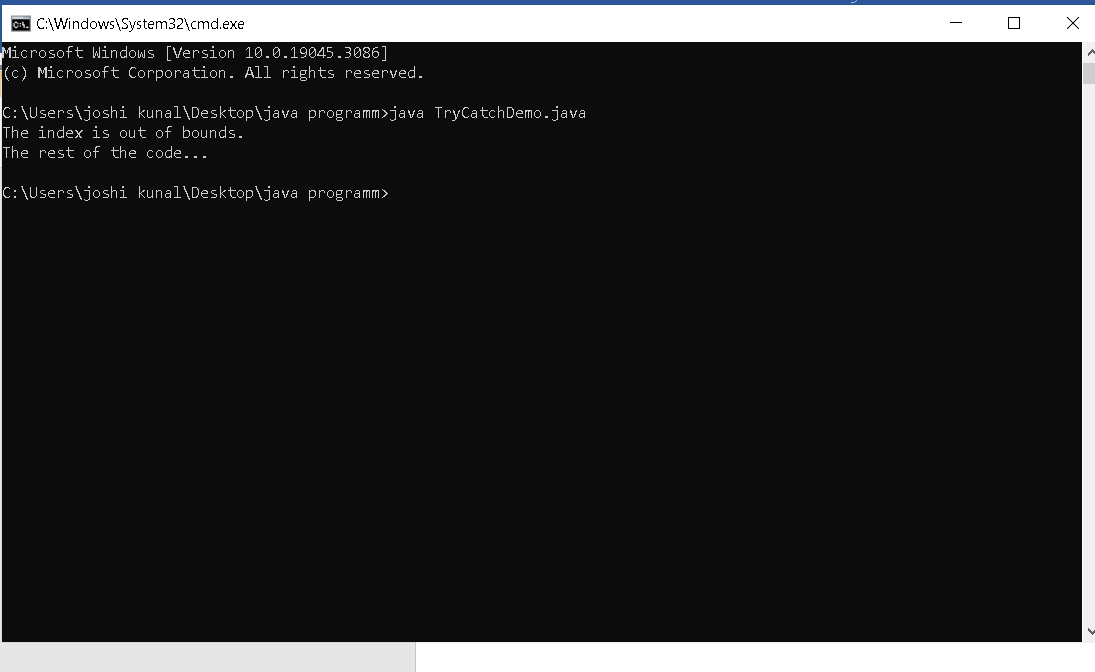
}

System.out.println("The rest of the code...");

}

}

Output:



1. WAP to demonstrate multiple catch blocks

Code:

public class MultipleCatchBlocks {

public static void main(String[] args) {

try {

int[] numbers = {1, 2, 3};

// This code may throw an exception.

System.out.println(numbers[4]);

// This code may also throw an exception.

String str = null;

System.out.println(str.length());

} catch (ArrayIndexOutOfBoundsException e) {

// This code will be executed if the first exception is thrown.

System.out.println("The index is out of bounds.");

} catch (NullPointerException e) {

// This code will be executed if the second exception is thrown.

System.out.println("The string is null.");

} catch (Exception e) {

// This code will be executed if any other exception is thrown.

System.out.println("An unknown exception occurred.");

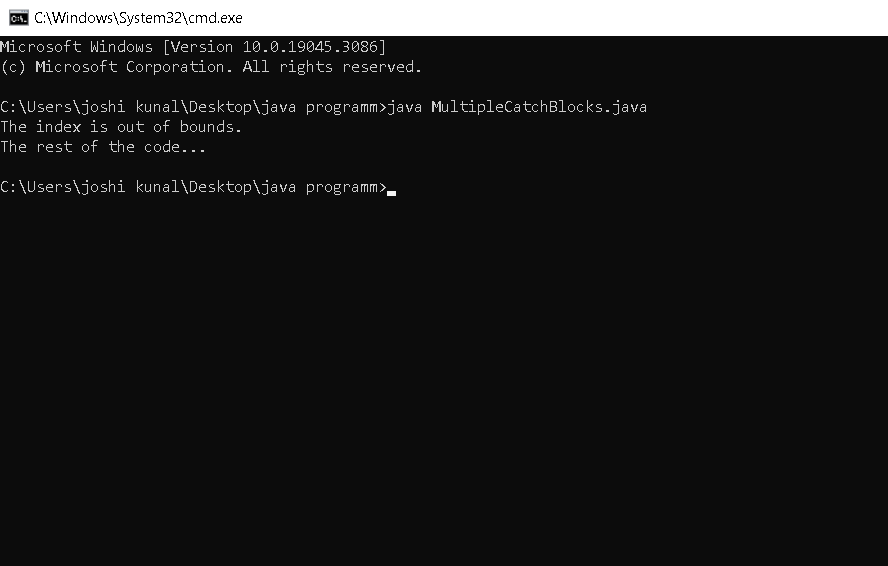
}

System.out.println("The rest of the code...");

}

}

Output:



1. WAP to create one thread by implementing Runnable interface in Class.

Code:

import java.util.concurrent.TimeUnit;

public class MyRunnable implements Runnable {

@Override

public void run() {

System.out.println("Thread started");

try {

TimeUnit.SECONDS.sleep(5);

} catch (InterruptedException e) {

e.printStackTrace();

}

System.out.println("Thread finished");

}

public static void main(String[] args) {

// Create a Thread object

Thread thread = new Thread(new MyRunnable());

// Start the thread

thread.start();

// Wait for the thread to finish

try {

thread.join();

} catch (InterruptedException e) {

e.printStackTrace();

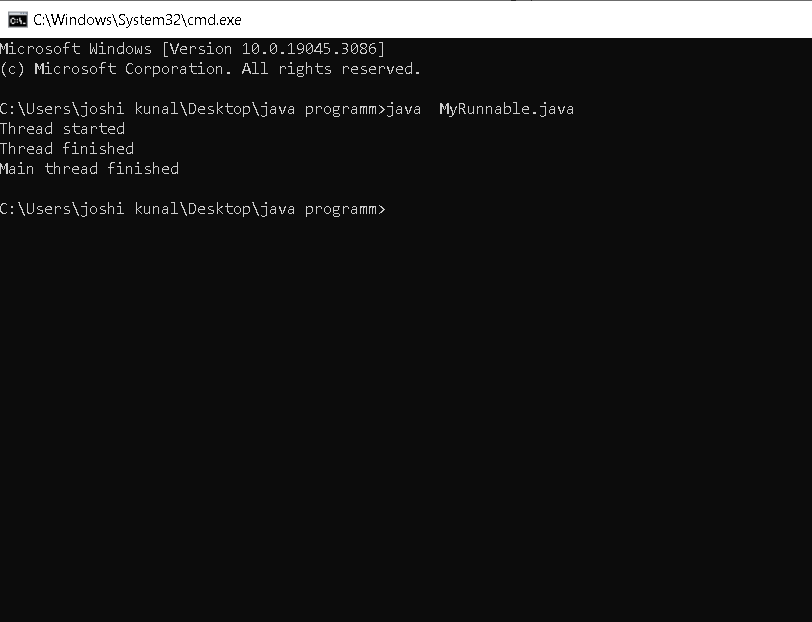
}

System.out.println("Main thread finished");

}

}

Output:



1. WAP to create one thread by extending Thread class in another

Code:

class MyThread extends Thread {

@Override

public void run() {

System.out.println("Thread started");

try {

Thread.sleep(5000);

} catch (InterruptedException e) {

e.printStackTrace();

}

System.out.println("Thread finished");

}

public static void main(String[] args) {

// Create a MyThread object

MyThread thread = new MyThread();

// Start the thread

thread.start();

// Do some other work in the main thread

System.out.println("Doing some work in the main thread");

// Wait for the thread to finish

try {

thread.join();

} catch (InterruptedException e) {

e.printStackTrace();

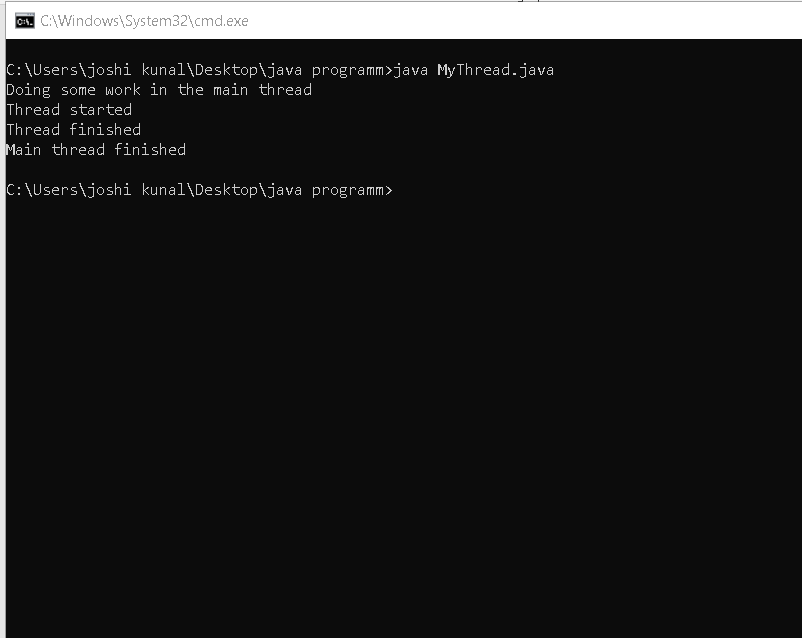
}

System.out.println("Main thread finished");

}

}

Output:



1. WAP to update specific array element by given element.

Code:

public class UpdateArrayElement {

public static void main(String[] args) {

// Create an array of integers

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

// Get the index of the element to be updated

int index = 2;

// Get the new element to be used for updating

int newElement = 10;

// Update the element at the specified index

arr[index] = newElement;

// Print the updated array

System.out.println("The updated array is:");

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

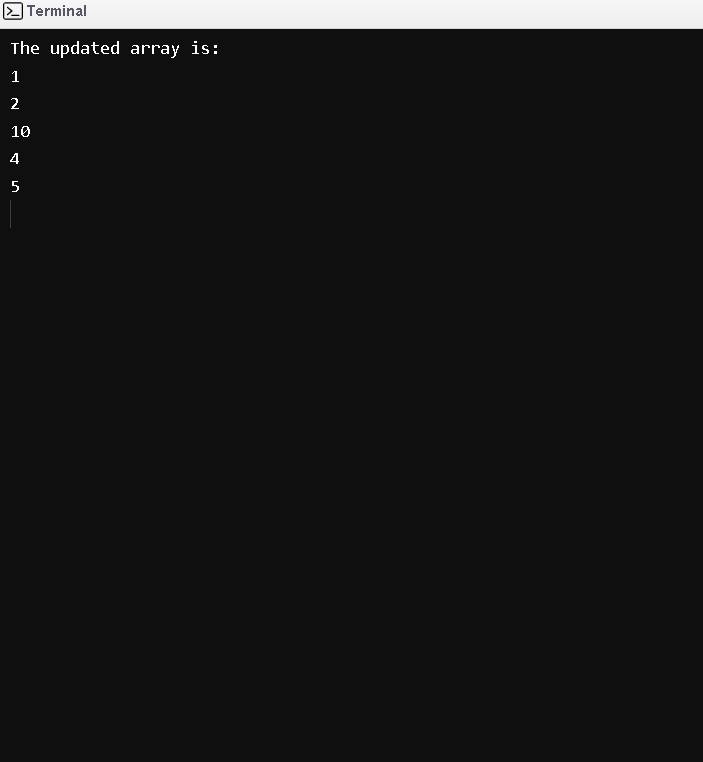
System.out.println(arr[i]);

}

}

}

Output:



1. WAP to remove the third element from a array list.

Code:

import java.util.ArrayList;

public class RemoveThirdElement {

public static void main(String[] args) {

ArrayList<String> list = new ArrayList<>();

list.add("First");

list.add("Second");

list.add("Third");

list.add("Fourth");

list.add("Fifth");

// Get the index of the third element

int thirdElementIndex = 2;

// Remove the third element from the list

list.remove(thirdElementIndex);

// Print the list

System.out.println(list);

}

}

Output:



1. WAP to Copy one array into another

Code:

public class CopyArray {

public static void main(String[] args) {

// Initialize the original array

int[] arr1 = {1, 2, 3, 4, 5};

// Create a new array to store the copy

int[] arr2 = new int[arr1.length];

// Copy the elements of arr1 into arr2

for (int i = 0; i < arr1.length; i++) {

arr2[i] = arr1[i];

}

// Display the original array

System.out.println("Elements of original array: ");

for (int i = 0; i < arr1.length; i++) {

System.out.print(arr1[i] + " ");

}

// Display the copied array

System.out.println();

System.out.println("Elements of copied array: ");

for (int i = 0; i < arr2.length; i++) {

System.out.print(arr2[i] + " ");

}

}

}

Output:



1. WAP to reverse an array of integer values.

Code:

import java.util.Arrays;

public class ReverseArray {

public static void main(String[] args) {

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

System.out.println("Original array: " + Arrays.toString(array));

// Reverse the array using in-place reversal.

for (int i = 0; i < array.length / 2; i++) {

int temp = array[i];

array[i] = array[array.length - i - 1];

array[array.length - i - 1] = temp;

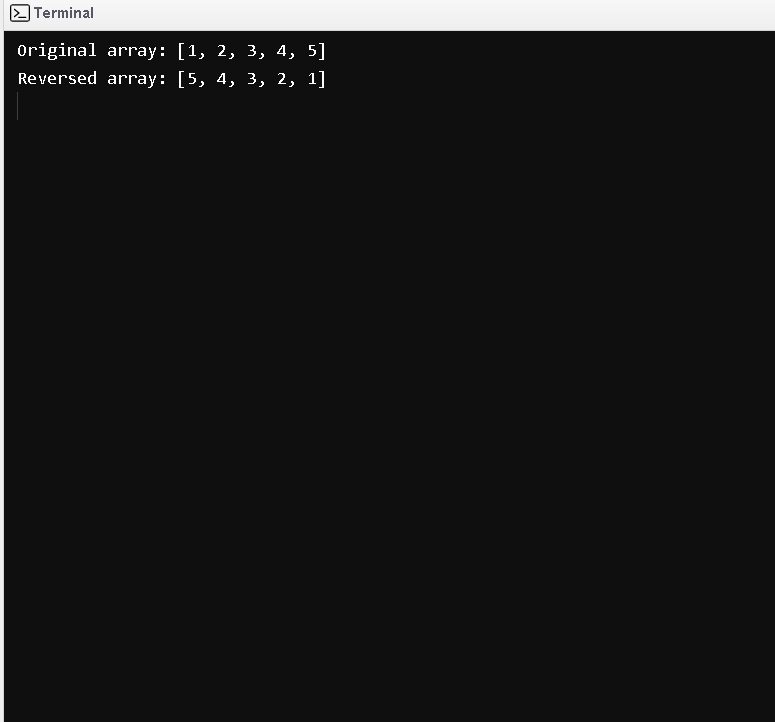
}

System.out.println("Reversed array: " + Arrays.toString(array));

}

}

Output:



1. WAP to find the second largest element in an array

Code:

import java.util.Arrays;

public class SecondLargestElement {

public static void main(String[] args) {

int[] arr = {10, 4, 2, 9, 7, 5, 3, 8, 6};

// Initialize two variables to store the largest and second largest elements

int largest = arr[0];

int secondLargest = arr[0];

// Iterate through the array and update the largest and second largest elements if necessary

for (int i = 1; i < arr.length; i++) {

if (arr[i] > largest) {

secondLargest = largest;

largest = arr[i];

} else if (arr[i] > secondLargest) {

secondLargest = arr[i];

}

}

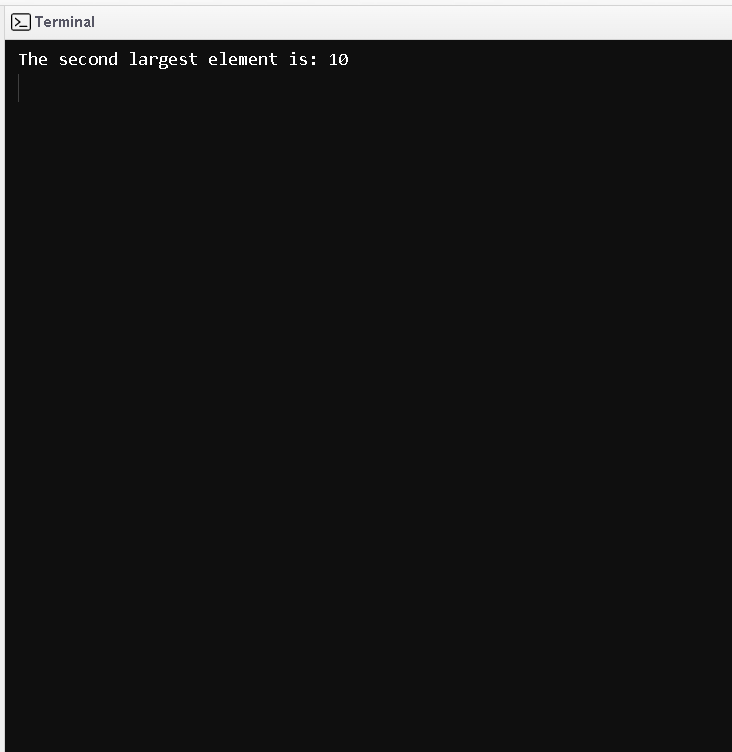
// Print the second largest element

System.out.println("The second largest element is: " + secondLargest);

}

}

Output:



1. W.A.J.P. Create an abstract class 'Parent' with a method 'message'. It has two subclasses each having a method with the same name 'message' that prints "This is first subclass" and "This is second subclass" respectively. Call the methods 'message' by creating an object for each subclass.\

Code:

abstract class Parent {

abstract void message();

}

class FirstSubclass extends Parent {

@Override

void message() {

System.out.println("This is first subclass");

}

}

class SecondSubclass extends Parent {

@Override

void message() {

System.out.println("This is second subclass");

}

}

public class Main {

public static void main(String[] args) {

Parent firstSubclass = new FirstSubclass();

firstSubclass.message();

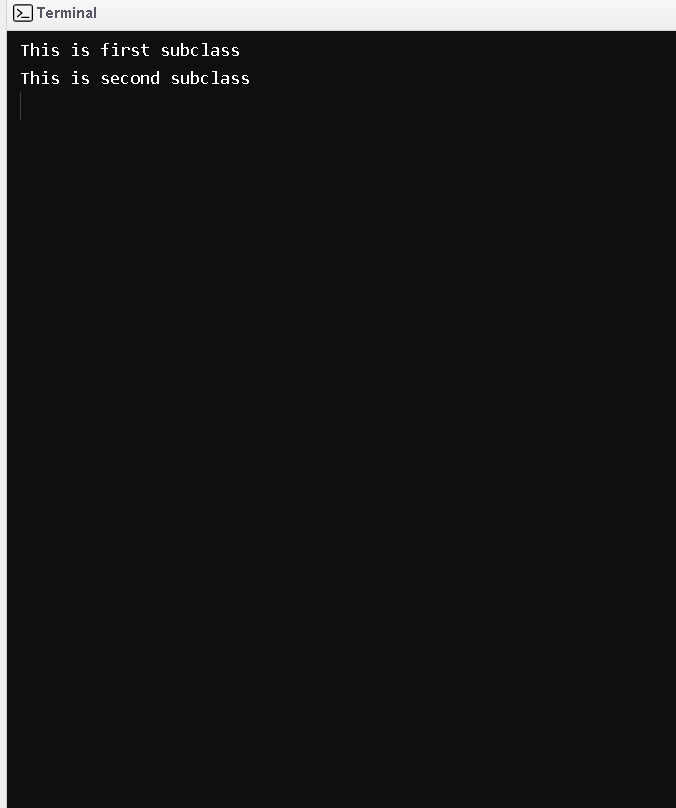
Parent secondSubclass = new SecondSubclass();

secondSubclass.message();

}

}

Output:



1. W.A.J.P. which will ask the user to enter his/her marks (out of 100). Define a method that will display grades according to the marks entered as below: Marks Grade 91-100 AA 81-90 AB 71-80 BB 61-70 BC 51-60 CD 41-50 DD <=40 Fail

Code:

import java.util.Scanner;

public class GradeCalculator {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.println("Enter your marks (out of 100): ");

int marks = scanner.nextInt();

String grade = getGrade(marks);

System.out.println("Your grade is " + grade);

}

private static String getGrade(int marks) {

String grade;

if (marks >= 91 && marks <= 100) {

grade = "AA";

} else if (marks >= 81 && marks <= 90) {

grade = "AB";

} else if (marks >= 71 && marks <= 80) {

grade = "BB";

} else if (marks >= 61 && marks <= 70) {

grade = "BC";

} else if (marks >= 51 && marks <= 60) {

grade = "CD";

} else if (marks >= 41 && marks <= 50) {

grade = "DD";

} else {

grade = "Fail in core java";

}

return grade;

}

}

Output:

