1. **Demonstrate a usage of package**

Write a Java program that generates and prints the first 'n' prime numbers using a single package named primepackage. Please follow the instructions below:

* Create a package named **primepackage**.
* Inside the primepackage package, create a class named **PrimeGenerator**.
* Inside the PrimeGenerator class, implement a static method **generatePrimes** that takes an integer 'n' as input and returns an array of the first 'n' prime numbers.

**PrimeGenerator.java**

**package** primepackage;

public class **PrimeGenerator**

{

// Static method to generate the first 'n' prime numbers

public static int[] **generatePrimes**(int n)

{

// An array to store the first 'n' prime numbers

int[] primes = new int[n];

int count = 0; // To count the number of primes found

int number = 2; // Starting point (first prime number)

while (count < n)

{

if (isPrime(number))

{

primes[count] = number;

count++;

}

number++;

}

return primes;

}

// Helper method to check if a number is prime

private static boolean isPrime(int num)

{

if (num <= 1)

{

return false; // Numbers less than or equal to 1 are not prime

}

for (int i = 2; i <= Math.sqrt(num); i++)

{

if (num % i == 0)

{

return false; // Divisible by i, hence not prime

}

}

return true; // Prime number

}

}

**Main.java**

**import primepackage.PrimeGenerator;**

import java.util.\*;

public class **Main**

{

public static void **main**(String[] args)

{

Scanner scanner = new Scanner(System.in);

// Taking input from the user for the number of primes to generate

System.out.print("Enter the number of prime numbers to generate: ");

int n = scanner.nextInt();

// Generate the first 'n' primes using the generatePrimes method

int[] primes = PrimeGenerator.generatePrimes(n);

// Printing the generated prime numbers

System.out.println("First " + n + " prime numbers are:");

for (int prime : primes)

{

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

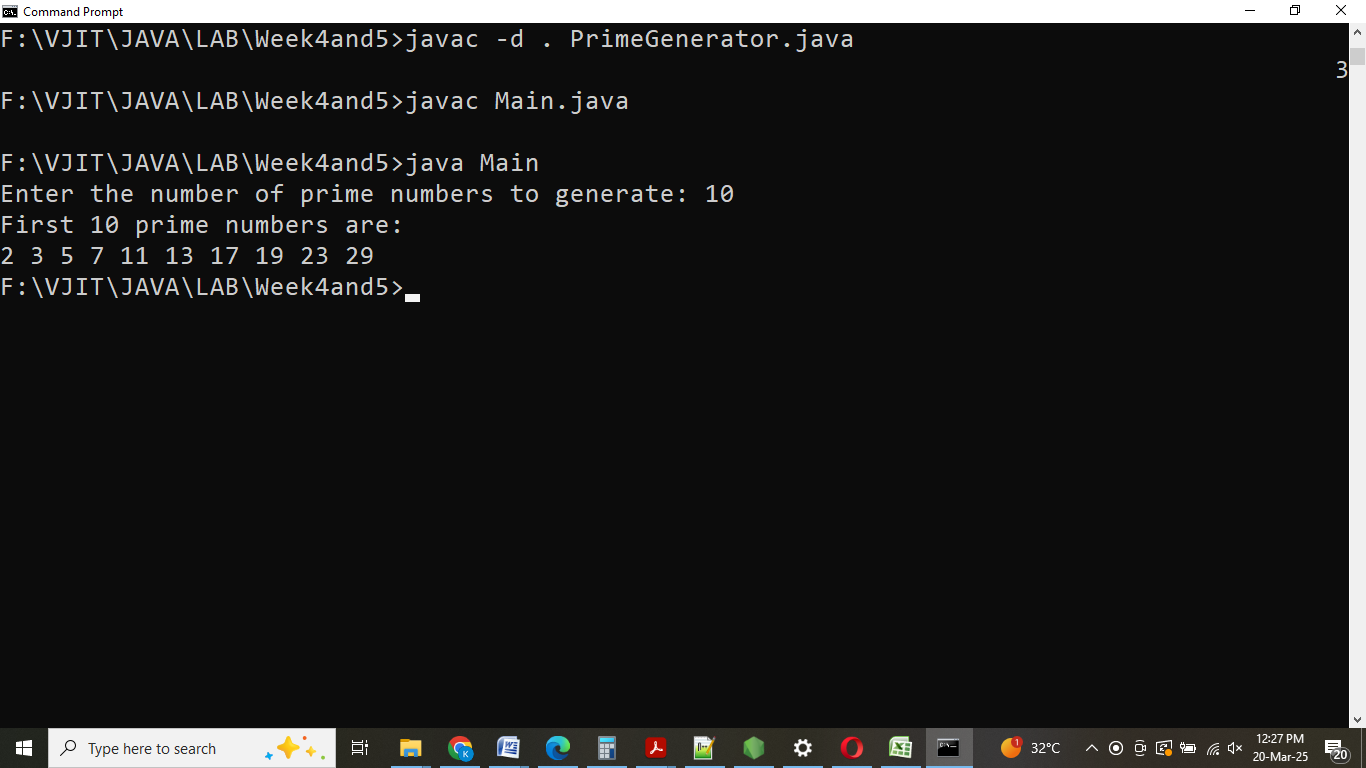
}

scanner.close();

}

}

**Output:**



1. **Multiple Catch blocks using command line arguments**

Write a Java Program to illustrate Multiple Catch blocks using command line argument. Refer to the sample test case provided to write the code.

**Input Format:**

The program takes two command-line arguments:

* The first argument is the numerator (an integer).
* The second argument is the denominator (an integer).

public class **PrintDivMain**

{

public static void **main**(String[] args)

{

**try**

{

//two command-line arguments converted into integer values

// Check if the command-line arguments are valid

if (args.length < 2)

{

System.out.println("Insufficient inputs");

return; // Exit if there are not enough arguments

}

// Convert the command-line arguments to integers

int numerator = Integer.parseInt(args[0]);

int denominator = Integer.parseInt(args[1]);

**try**

{

//perform division operation and catch exception

if (denominator == 0)

{

// Throw an exception if dividing by zero

throw new ArithmeticException(); }

// Perform the division and print the result

int result = numerator / denominator;

System.out.println("Division: " + result);

}

**catch**(ArithmeticException e)

{

// Handle division by zero

System.out.println("Cannot divide with zero");

}

}

**catch**(NumberFormatException e)

{

// Handle invalid input (non-integer type)

System.out.println("Input is not of integer type");

}

**catch**(Exception e)

{

// Catch any other unexpected exceptions

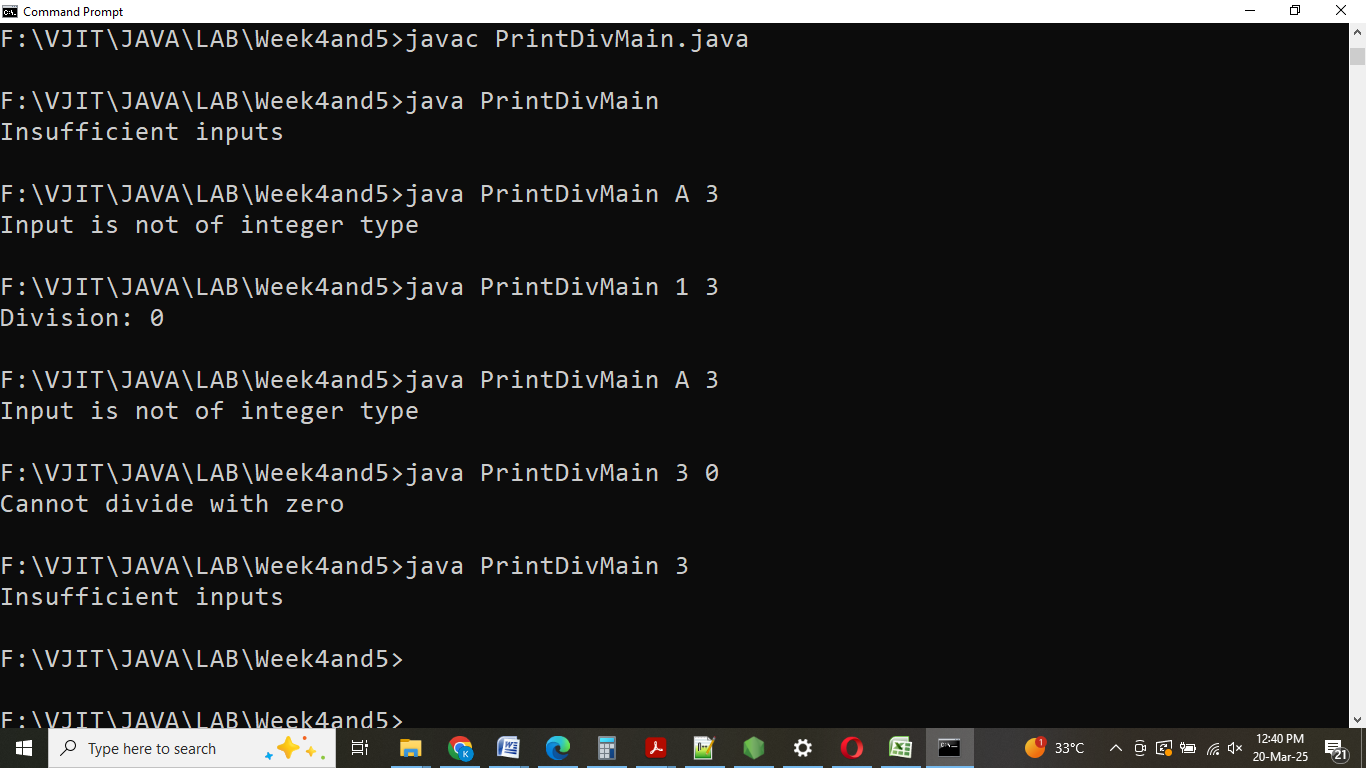
System.out.println("An unexpected error occurred");

}

}

}

**Output:**



1. **User Defined Exceptions**

Write a Java program that checks if a person is eligible to vote based on their age.

* If the age is below 18 or above 60, throw a custom exception with the message: "Age must be between 18 and 60"
* If the age is 18 or above, but below 60, print "eligible to vote"
* Handle any invalid input gracefully (e.g., non-integer inputs).

You need to create a custom exception class named **VotingEligibilityException** to handle the age validation.

import java.util.Scanner;

class **VotingEligibilityException** extends **Exception**

{

// Constructor to pass the error message to the base Exception class

public **VotingEligibilityException**(String message)

{

super(message);

}

}

public class **VotingEligibility**

{

public static void **main**(String[] args)

{

Scanner scanner = new Scanner(System.in);

try

{

// Prompt user to enter age

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

int age = scanner.nextInt(); // Read user input

// Check if the age is in the valid voting range

if (age < 18 || age > 60)

{

**throw** new VotingEligibilityException("Age must be between 18 and 60");

}

else

{

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

}

}

**catch** (VotingEligibilityException e)

{

// Handle custom exception (if age is outside valid range)

System.out.println(e.getMessage());

}

**catch** (Exception e)

{

// Handle invalid input (e.g., non-integer input)

System.out.println("Invalid input");

}

**finally**

{

// Close the scanner

scanner.close();

}

}

}

**Output:**

