

Table of Contents



Program in C

Program in Java

Write a program that checks if a given number is prime.

Prime number: A prime number is a natural number greater than 1 that has no positive divisors other than 1 and itself.

Program In C



```
#include <stdio.h>
#include <stdbool.h>

int main() {
    int num;
    bool isPrime = true;

    printf("Enter a positive integer: ");
    scanf("%d", &num);


    if (num <= 1) {
        isPrime = false;
    } else {
        for (int i = 2; i * i <= num; i++) {
            if (num % i == 0) {
                isPrime = false;
                break;
            }
        }
    }
}
```

```
    }  
}  
  
if (isPrime) {  
    printf("%d is a prime number.\n", num);  
} else {  
    printf("%d is not a prime number.\n", num);  
}  
  
return 0;  
}
```

Explanation:

- The main function prompts the user to enter a positive integer, reads it from the standard input using scanf, and stores it in the num variable.
- We also initialize a boolean variable isPrime to true, assuming initially that the number is prime.
- We then perform the prime check. If the number is less than or equal to 1, we set isPrime to false since prime numbers are greater than 1. Otherwise, we iterate through the numbers from 2 to the square root of num using a for loop. For each iteration, we check if the number is divisible by i using the modulo operator (%). If it is divisible, we set isPrime to false and break out of the loop since we have found a divisor.
- Finally, we use an if-else statement to display an appropriate message based on the value of isPrime.

Output:

Output 

```
Enter a positive integer: 5  
5 is a prime number.
```

Program In Java

java 

```
import java.util.Scanner;  
  
public class PrimeNumberJava {  
    public static void main(String[] args) {  
        int num;  
        boolean isPrime = true;  
  
        System.out.print("Enter a positive integer: ");  
        Scanner scanner = new Scanner(System.in);  
        num = scanner.nextInt();  
  
        if (num <= 1) {  
            isPrime = false;  
        } else {  
            for (int i = 2; i * i <= num; i++) {  
                if (num % i == 0) {  
                    isPrime = false;  
                    break;  
                }  
            }  
        }  
    }  
}
```

```
        if (isPrime) {
            System.out.printf("%d is a prime number.\n", num);
        } else {
            System.out.printf("%d is not a prime number.\n", num);
        }

        scanner.close();
    }
}
```

Explanation:

1. `import java.util.Scanner;`: This line imports the Scanner class from the java.util package, allowing us to read user input from the console.
2. `public class PrimeNumberJava:` This line defines a public class named PrimeNumberJava.
3. `public static void main(String[] args):` This line declares the main method, which is the entry point of the program.
4. `int num;`: This line declares an integer variable num to store the user input.
5. `boolean isPrime = true;`: This line declares a boolean variable isPrime and initializes it to true. It will be used to determine whether the number is prime or not.
6. `System.out.print("Enter a positive integer: ");`: This line prompts the user to enter a positive integer.
7. `Scanner scanner = new Scanner(System.in);`: This line creates a Scanner object named scanner to read user input from the console.
8. `num = scanner.nextInt();`: This line reads an integer from the user and assigns it to the variable num.
9. The code block starting with `if (num <= 1) {` checks if the entered number is less than or equal to 1. If so, it sets isPrime to false since prime numbers are greater than 1.

10. If the number is greater than 1, a for loop is used to check for factors of the number. The loop starts with `int i = 2`, and it continues as long as `i * i <= num`. The condition `i * i <= num` optimizes the loop by reducing the number of iterations needed to check for factors.
11. Inside the loop, the code checks if `num` is divisible evenly by `i` using the condition `num % i == 0`. If it is, the number is not prime, and `isPrime` is set to `false`. The `break` statement is used to exit the loop early since no further checking is necessary.
12. After the loop, an if-else statement is used to print whether the number is prime or not based on the value of `isPrime`.
13. `scanner.close();`: This line closes the `Scanner` object to release system resources.

Output

```
Enter a positive integer: 4
4 is not a prime number.
```

Related posts:

1. Programming examples
2. Program to prints even numbers from 1 to 20
3. Program to calculate sum of all numbers from 1 to 100.
4. Program to get factorial of a number
5. Program to get Fibonacci sequence
6. Program to get multiplication table
7. Program to find largest element in an array
8. Program to prints reverse of a string
9. Program to calculates sum of all elements in a list
10. Program determines integer is positive, negative, or zero

11. Program to find largest among three numbers using conditional statements.
12. Program determines it is a leap year or not
13. Program to determines even or odd
14. Program to calculate student exam grade
15. Program determines character is a vowel or consonant
16. Program to determines product is positive or negative
17. Program to determine divisible by both 5 and 7
18. Program to determines equilateral, isosceles, or scalene triangle
19. Programme to check if number is inside range
20. Function to calculate the factorial
21. Write a function to detect palindromes in strings
22. Write a function to find the greatest common divisor of two numbers
23. Program to calculate the area of different geometric shapes
24. try-catch block in C++