Checking Whether a Number is Even or Odd Using Java

K. NAGA VIHARI

Guest Faculty, Department of Statistics, Sri Venkateswara University, Tirupati, AP, India.

Abstract— Determining whether a number is even or odd is an important subject in programming and mathematics. This article describes a simple and efficient Java program for determining if a given number is even or odd. The software uses the modulus operator (%) to calculate the remainder when an integer is split by two. If the remainder is 0, the number is considered even; otherwise, it is odd. The article includes a step-by-step description of the code, from reading input with a Scanner object to implementing the logic for calculating parity. The program's efficiency makes it appropriate for a variety of applications, including mathematical methods, resource allocation, and game logic. By learning this fundamental procedure, programmers may lay a solid basis for more difficult tasks in Java and other programming languages.

Index Terms- Even and Odd Numbers, Java Programming, Parity Check

I. INTRODUCTION

Determining whether a number is even or odd is one of the most basic operations in programming. An even number is divisible by 2 without leaving a remainder, while an odd number leaves a remainder when divided by 2. In this article, we will explore how to create a simple Java program to check whether a given number is even or odd.

Understanding Even and Odd Numbers

- Even Numbers: Numbers like 2, 4, 6, 8, and so on are even because they can be evenly divided by 2. Mathematically, a number n is even (i.e., if it leaves no remainder when divided by 2).
- Odd Numbers: Numbers like 1, 3, 5, 7, etc., are odd because dividing them by 2 leaves a remainder of 1. Mathematically, a number n is odd The modulus operator (%) is commonly used in programming to find the remainder of a division operation. This operator plays a crucial role in our program for determining whether a number is even or odd.

Java Program Implementation

Below is the Java code that checks if a number is even or odd:

java

Copy code

```
import java.util.Scanner;

public class EvenOddCheck {
    public static void main(String[] args) {
        // Create a Scanner object to read input
        Scanner scanner = new Scanner(System.in);

System.out.print("Enter a number: ");
int number = scanner.nextInt(); // Read the number

        // Check if the number is even or odd
        if (number % 2 == 0) {
System.out.println(number + " is even.");
        } else {
System.out.println(number + " is odd.");
        }

        // Close the scanner
scanner.close();
    }
}
```

Explanation of the Code

1. Reading Input: The application invites users to input a number. This is accomplished by utilizing a Scanner object, which allows us to read information from the console. The entered number is saved in the variable number.
2. Checking Even or Odd: The rationale is in the if-else section. If (number%2 == 0): This condition determines whether the residue when dividing an integer by two is zero. If it is, the number is even, and the software outputs "even." If the condition is false (i.e., the residual is not zero), the number will be shown as odd.
3. Close the Scanner: The application closes the scanner to free up resources.

Sample Output

Here’s a sample run of the program:

Copy code

```
Enter a number: 45
45 is odd.
```

Another example:

Copy code

```
Enter a number: 16
16 is even.
```

This program's simplicity stems from its use of the modulus operator, which allows us to compute the parity (evenness or oddness) of a number in a single line of code.

Real-world applications:

- While determining whether a number is even or odd may appear easy, this idea has several useful applications:
- Mathematical algorithms: Many algorithms, particularly those involving sequences and patterns, are based on the contrast between even and odd integers.
- Some systems utilize even and odd integers as resource IDs, resulting in differing allocations.
- Even and odd logic are commonly employed in games to assign turns or roles.

CONCLUSION

This article shows how to create a basic Java program that checks if a number is even or odd. Understanding this fundamental procedure is a necessary foundation for more sophisticated programming jobs. With just a few lines of code, you can detect if a number is even or odd, and this knowledge serves as the foundation for researching more sophisticated topics in mathematics and computer science.

REFERENCES

[1] GeeksforGeeks: Even and Odd Number in Java

[2] JavaTPoint: Java Program to Check Even or Odd Number

[3] Baeldung: Java Modulus Operator