OPERATOR OVERLOADING

AIM:

To create operator overloading for all the arithmatic operators.

ALGORITHM:

· Define a class Number with an integer data member.

· Initialize the data member using a parameterized constructor.

· Overload the arithmetic operators +, -, *, /, and %.

· Overload the == operator for comparison.

· Define a function to display the value.

· Create two objects and initialize them with values.

· Perform arithmetic operations using overloaded operators.

· Display the result after each operation.

· Compare the two objects and display the result.

PROGRAM:

```
/*
*  Program to demonstrate all arithmetic operators overloading in C++
*  Author  : MUTHUGANESH S
*  Date    : 27/1/2026
*  Filename: Overloading.cpp
*  retval  : void
*/

#include <iostream>
using namespace std;

class Number{

private:
    int Value;

public:
    Number(int v){

        Value = v;
    }

    // Overloading + operator
    void operator +(const Number &num){
```

```
        Value = this->Value + num.Value;
    }

    // Overloading - operator
    void operator -(const Number &num){

        Value = this->Value - num.Value;
    }

    // Overloading * operator
    void operator *(const Number &num){

        Value = this->Value * num.Value;
    }

    // Overloading / operator
    void operator /(const Number &num){

        Value = this->Value / num.Value;
    }

    // Overloading % operator
    void operator %(const Number &num){

        Value = this->Value % num.Value;
    }

    Number operator ==(const Number &num){

        return Number(this->Value == num.Value);
    }

    void display(){
        cout << "Value: " << Value << endl;
    }
};


int main(){

    Number num1(20);
    Number num2(10);

    num1 + num2;
    cout << "After Addition: ";
    num1.display();

    num1 - num2;
    cout << "After Subtraction: ";
    num1.display();

    num1 * num2;
    cout << "After Multiplication: ";
```

```
    num1.display();

    num1 / num2;
    cout << "After Division: ";
    num1.display();

    num1 % num2;
    cout << "After Modulus: ";
    num1.display();

    Number result = num1 == num2;
    cout << "After Equality Check: ";
    result.display();

    return 0;
}
```

OUTPUT: