# BANGLADESH UNIVERSITY OF BUSINESS AND TECHNOLOGY (BUBT)


## Lab Report

Course Code : CSE 324

Course Title : Compiler Design Lab

Date of Submission: February 11, 2024


Submitted By

Name : Aktaruzzaman

ID : 21222203031

Intake : 41

Section : 1

Submitted To

Ms. Adeeba Anis
Lecturer

Department of Computer Science & Engineering

Bangladesh University of Business and Technology (BUBT)

**Experiment No:** 4

**Experiment Name:** Program for Identifying operator

## Problem Structure

The objective of this experiment is to create a C++ program that can identify various operators entered by the user and categorize them based on their type (e.g., arithmetic, relational, logical, etc.). The program should take input from the user and determine whether the entered string corresponds to an operator, and if so, classify it accordingly.

## Procedure

- Define a function checkOperator that takes a string as input.
- Inside the function, use a series of if-else statements to compare the input string with predefined operator strings.
- Based on the comparison results, print out the corresponding operator type.
- In the main function, prompt the user to input an operator string.
- Call the checkOperator function with the user input.
- Display the classification of the entered operator.

## Code

Start here | main.cpp

```cpp
#include<iostream>
using namespace std;

void checkOperator(string str){
    if(str == "+"){
        cout << "Addition, Arithmetic  Operator" << endl;
    }else if(str == "-"){
         cout << "Subtraction, Arithmetic  Operator" << endl;
    }else if(str == "*"){
         cout << "Multiplication, Arithmetic  Operator" << endl;
    }else if(str == "/"){
         cout << "Division, Arithmetic  Operator" << endl;
    }else if(str == "%"){
         cout << "Modulus, Arithmetic  Operator" << endl;
    }else  if(str == ">"){
        cout << "Greater than, Relational Operator" << endl;
    }else if(str == "<"){
         cout << "Less than, Relational Operator" << endl;
    }else if(str == ">="){
         cout << "Greater than equal, Relational Operator" << endl;
    }else if(str == "<="){
         cout << "Less than equal, Relational Operator" << endl;
    }else if(str == "!="){
         cout << "Not Equal, Relational Operator" << endl;
    }else if(str == "=="){
         cout << "Equal Equal, Relational Operator" << endl;
    }else if(str == "="){
        cout << "Equal, Assignment Operator" << endl;
    }else if(str == "+="){
        cout << "Additional assignment, Assignment Operator" << endl;
    }else if(str == "-="){
        cout << "Subtraction assignment, Assignment Operator" << endl;
    }else if(str == "*="){
        cout << "Multiplication assignment, Assignment Operator" << endl;
    }else if(str == "/="){
        cout << "Division assignment, Assignment Operator" << endl;
    }else if(str == "%="){
        cout << "Modulus assignment, Assignment Operator" << endl;
    }else if(str == "%="){
        cout << "Modulus assignment, Assignment Operator" << endl;
    }else if(str == "&&"){
```

```
42          cout << "and, Logical Operator" << endl;
43      }else if(str == "||"){
44          cout << "or, Logical Operator" << endl;
45      }else if(str == "!"){
46          cout << "not, Logical Operator" << endl;
47      }else if(str == "&"){
48          cout << "Bitwise and, Bitwise Operator" << endl;
49      }else if(str == "|"){
50          cout << "Bitwise or, Bitwise Operator" << endl;
51      }else if(str == "^"){
52          cout << "Bitwise or, Bitwise Operator" << endl;
53      }else if(str == "~"){
54          cout << "Binary One's Complement, Bitwise Operator" << endl;
55      }else if(str == "<<"){
56          cout << "Binary Shift Left, Bitwise Operator" << endl;
57      }else if(str == ">>"){
58          cout << "Binary right shift, Bitwise Operator" << endl;
59      }else if(str == "++"){
60          cout << "Increment, Unary Operator" << endl;
61      }else if(str == "--"){
62          cout << "Decrement, Unary Operator" << endl;
63      }else if(str == "?:"){
64          cout << "Ternary, Ternary Operator" << endl;
65      }else{
66          cout << "Not an operator";
67      }
68  }
69
70  int  main(){
71      string input;
72      cout << "Please Enter an Operator: " ;
73      getline(cin, input);
74
75      checkOperator(input);
76
77  }
```

## Input and Output

```
coderaktar@root: ~/Desktop/Identify Operator
coderaktar@root:~/Desktop/Identify Operator$ ./main
Please Enter an Operator: +=
Additional assignment, Assignment Operator
coderaktar@root:~/Desktop/Identify Operator$ ./main
Please Enter an Operator: +-
Not an operator
coderaktar@root:~/Desktop/Identify Operator$ ./main
Please Enter an Operator: ==
Equal Equal, Relational Operator
coderaktar@root:~/Desktop/Identify Operator$ ./main
Please Enter an Operator: &
Bitwise and, Bitwise Operator
coderaktar@root:~/Desktop/Identify Operator$
```

## Conclusion

The program successfully identifies the type of operator entered by the user and categorizes it into different groups based on its functionality. It provides a simple and efficient way to understand the purpose of various operators in C++ programming.