BANGLADESH UNIVERSITY OF BUSINESS AND TECHNOLOGY (BUBT)

Lab Report

Course Code : CSE 324

Course Title : Compiler Design Lab

Date of Submission: January 28, 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: 2

Experiment Name: Valid Identifier Detection.

Problem Structure

The objective of this experiment is to design and implement a C/C++ program that determines whether a given string is a valid identifier or not. An identifier is considered valid if it adheres to the standard conventions of identifiers in programming languages. The program should provide clear feedback about the validity of the entered identifier.

Additional Constraints:

- The identifier must start with either an underscore (_) or a letter (either lowercase or uppercase).
- Subsequent characters in the identifier can include underscores, lowercase letters, uppercase letters, and digits.
- Special characters, including points, are not allowed in the identifier.

Procedure

1. Input Collection:
    - The program prompts the user to input an identifier using the `getline` function.
2. Identifier Validation:
    - The program checks the first character of the entered identifier to determine its validity. A valid identifier can start with an underscore (`_`), a lowercase letter (`a` to `z`), or an uppercase letter (`A` to `Z`).
    - For the subsequent characters, the program iterates through the identifier using a `for` loop and checks each character's conformity to the rules of a valid identifier. Valid characters include underscores, lowercase letters, uppercase letters, and digits.
3. Output Presentation:
    - The program presents the result of the validation by displaying whether the entered string is a valid identifier or not.

Code

lab-02.cpp

```cpp
#include<iostream>
using namespace std;
int main()
{
    string identefier;
    cout << "Please Enter Your Identifier: ";
    getline(cin, identefier);
    bool flag = true;

    if(identefier[0] == '_' ||
            (identefier[0] >=  'a' && identefier[0] <= 'z')
            || (identefier[0] >=  'A' && identefier[0] <= 'Z'))
    {
        for (size_t i = 1; i < identefier.length(); i++)
        {
            if(
                !(identefier[i] == '_' ||
                  (identefier[i] >= 'a' && identefier[i] <= 'z') ||
                  (identefier[i] >= 'A' && identefier[i] <= 'Z') ||
                  isdigit(identefier[i])))
            {
                flag = false;
                break;
            }
        }

        if(flag == false)
        {
            cout << identefier << " is Not Valid Identifier." << endl;
        }
        else
        {
            cout << identefier << " Valid Identifier." << endl;
        }

    }
    else
    {
        cout << identefier << " is Not Valid Identifier." << endl;
    }
}
```

Input and Output

```
coderaktar@root:~/Desktop/Check Valid Identifier Or Not$ ./lab-02
Please Enter Your Identifier: validOrNot
validOrNot is Valid Identifier.
coderaktar@root:~/Desktop/Check Valid Identifier Or Not$ ./lab-02
Please Enter Your Identifier: valid||Not
valid||Not is Not Valid Identifier.
coderaktar@root:~/Desktop/Check Valid Identifier Or Not$ ./lab-02
Please Enter Your Identifier: 1sItvalid
1sItvalid is Not Valid Identifier.
coderaktar@root:~/Desktop/Check Valid Identifier Or Not$ ./lab-02
Please Enter Your Identifier: _1sItValid
_1sItValid is Valid Identifier.
coderaktar@root:~/Desktop/Check Valid Identifier Or Not$ ./lab-02
Please Enter Your Identifier: is_it_valid
is_it_valid is Valid Identifier.
```

Conclusion

The experiment successfully addressed the task of validating identifiers based on standard conventions. The C/C++ program provides a user-friendly interface for entering identifiers and delivers accurate feedback on their validity. The implementation utilizes logical conditions and iteration to ensure comprehensive validation, making it a valuable exercise for understanding identifier validation in compiler design.