BANGLADESH UNIVERSITY OF BUSINESS AND TECHNOLOGY (BUBT)

Lab Report

Course Code : CSE 324

Course Title : Compiler Design Lab

Date of Submission: February 04, 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: 3

Experiment Name: Identifying Valid C++ Keywords

Problem Structure

The objective of this experiment is to design a C++ program that determines whether a given string is a valid C++ keyword or not. C++ keywords are reserved words in the language, and any attempt to use them as identifiers may result in a compilation error. The program should take a user-inputted string and check whether it corresponds to a valid C++ keyword.

Procedure

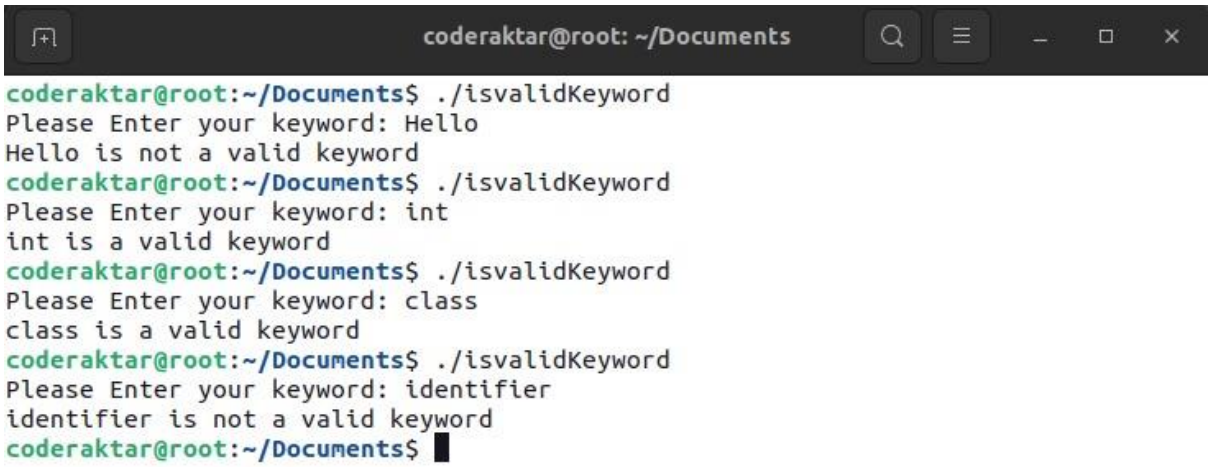
- **Keyword Array:** A list of valid C++ keywords is predefined in an array within the program.
- **User Input:** The program prompts the user to enter a keyword for validation.
- **Validation Function:** A function (isValidKeyword) iterates through the array of keywords, comparing each one with the user-inputted string.
- **Output:** The program prints whether the entered string is a valid C++ keyword or not.

Code

here × *valid_keyword.cpp ×

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

bool isValidKeyword(string str){
    string keywords[] = {
        "alignas", "alignof", "and", "and_eq", "asm", "auto", "bitand", "bitor", "bool", "break",
        "case", "catch", "char", "char16_t", "char32_t", "class", "compl", "const", "constexpr", "const_cast",
        "continue", "decltype", "default", "delete", "do", "double", "dynamic_cast", "else", "enum", "explicit",
        "export", "extern", "false", "float", "for", "friend", "goto", "if", "inline", "int",
        "long", "mutable", "namespace", "new", "noexcept", "not", "not_eq", "nullptr", "operator", "or",
        "or_eq", "private", "protected", "public", "register", "reinterpret_cast", "return", "short", "signed", "sizeof",
        "static", "static_assert", "static_cast", "struct", "switch", "template", "this", "thread_local", "throw", "true",
        "try", "typedef", "typeid", "typename", "union", "unsigned", "using", "virtual", "void", "volatile",
        "wchar_t", "while", "xor", "xor_eq"
    };

    for(string keyword: keywords){
        if(keyword == str){
            return true;
        }
    }

    return false;
}

int main(){
    string checkKeyword;
    cout << "Please Enter your keyword: ";
    getline(cin,checkKeyword);

    if(isValidKeyword(checkKeyword)){
        cout << checkKeyword << " is a valid keyword" << endl;
    }else{
       cout << checkKeyword << " is not a valid keyword" << endl;
    }
}
```

Input and Output



Conclusion

This program successfully checks whether a given string is a valid C++ keyword. It employs a predefined list of keywords and utilizes a function to perform the validation. The user-friendly interface enhances ease of use.