Bangladesh University of Business and Technology

(BUBT)

Lab Report

Course Title : Compiler Design Lab

Course Code : CSE 224

Lab report : 02

Submitted By:

Name : Shagor Robidas

ID No : 20234203067

Intake : 45

Section : 2

Program : B.Sc. Engg in CSE

Submitted To

Name : Md. Saddam Hossain.
(Assistant Professor)
Department of : CSE

Bangladesh University of Business & Technology

22/05/2025

Date of Submission :

Signature of Teacher

Problem no 01: Find Tokenization

Software Requirements: Code Blocks

Source Code:

```
#include <iostream>
#include <string>

using namespace std;

bool keyword(const string& s) {
    return s == "int" || s == "float" || s == "return" || s == "if" || s == "else";
}

bool number(const string& s) {
    for (char c : s)
        if (c < '0' || c > '9')
            return false;
    return !s.empty();
}

bool identifier(const string& s) {
    if (s.empty()) return false;
    if (!((s[0] >= 'a' && s[0] <= 'z') || (s[0] >= 'A' && s[0] <= 'Z') || s[0] == '_'))
        return false;
    for (int i = 1; i < (int)s.size(); i++) {
        if (!((s[i] >= 'a' && s[i] <= 'z') || (s[i] >= 'A' && s[i] <= 'Z') ||
              (s[i] >= '0' && s[i] <= '9') || s[i] == '_'))
            return false;
    }
    return true;
}

int main() {
    string line;
    cout << "Enter code (tokens separated by space):\n";
    getline(cin, line);

    string token = "";
    for (char ch : line) {
        if (ch == ' ') {
            if (!token.empty()) {
                if (keyword(token)) cout << token << " : Keyword\n";
```

```
            else if (number(token)) cout << token << " : Number\n";
            else if (identifier(token)) cout << token << " : Identifier\n";
            else cout << token << " : Unknown\n";
            token = "";
        }
    } else {
        token += ch;
    }
}

if (!token.empty()) {
    if (keyword(token)) cout << token << " : Keyword\n";
    else if (number(token)) cout << token << " : Number\n";
    else if (identifier(token)) cout << token << " : Identifier\n";
    else cout << token << " : Unknown\n";
}

return 0;
}
```

Output:

```
Output

Enter code (tokens separated by space):
int 1 the shagor main
int : Keyword
1 : Number
the : Identifier
shagor : Identifier
main : Identifier


=== Code Execution Successful ===
```

Conclusion: In problem 1 we learn how to Tokenize any Expresion.

Problem no 02: Detection and removal of single line and multi-line

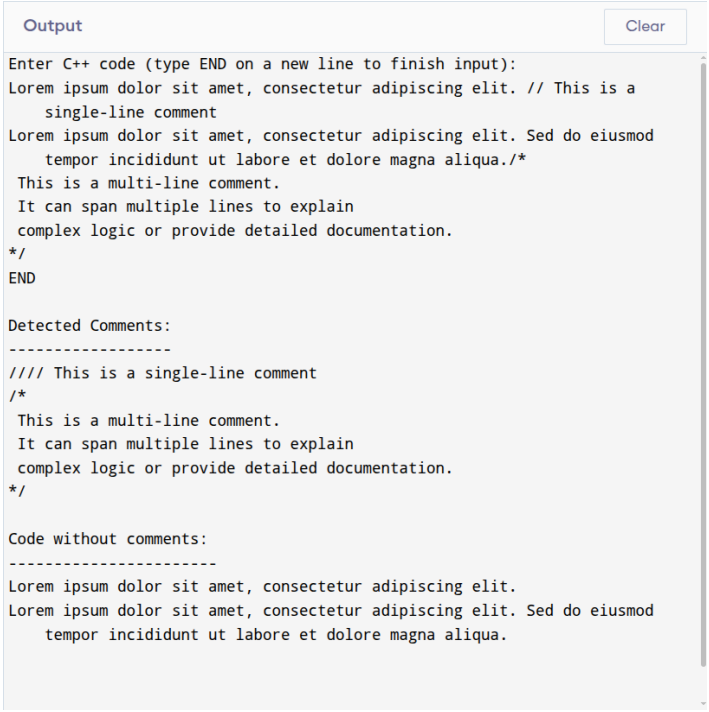
Software Requirements: Code Blocks

Source Code:

```
#include <iostream>
#include <string>

using namespace std;

int main() {
    cout << "Enter C++ code (type END on a new line to finish input):\n";

    string line, code;
    while (getline(cin, line)) {
        if (line == "END") break;
        code += line + '\n';
    }

    string result;
    bool comment = false;

    cout << "\nDetected Comments:\n";
    cout << "------------------\n";

    for (size_t i = 0; i < code.length(); ++i) {
        if (!comment && code[i] == '/' && i + 1 < code.length()) {
            if (code[i + 1] == '/') {
                cout << "//";
                while (i < code.length() && code[i] != '\n') {
                    cout << code[i];
                    i++;
                }
                cout << endl;
                result += '\n';
            }
            else if (code[i + 1] == '*') {
                comment = true;
                cout << "/*";
                i += 2;
                while (i + 1 < code.length() && !(code[i] == '*' && code[i + 1] == '/')) {
```

```
                cout << code[i];
                i++;
            }
            cout << "*/" << endl;
            i++;
            comment = false;
        }
        else {
            result += code[i];
        }
    }
    else {
        if (!comment)
            result += code[i];
    }
}

    cout << "\nCode without comments:\n";
    cout << "----------------------\n";
    cout << result << endl;

    return 0;
}
```

Output:



Conclusion: In problem 2 we learn how to detect and remove single line and multi-line comments**.**

Problem no 03: Remove extra white space

Software Requirements: Code Blocks

Source Code:

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

string removeSpaces(string s) {
    string result = "";
    bool space = false;

    for (char c : s) {
        if (c == ' ') {
            if (!space) {
                result += ' ';
                space = true;
            }
        } else {
            result += c;
            space = false;
        }
    }

    if (!result.empty() && result[0] == ' ')
        result.erase(0, 1);


    if (!result.empty() && result[result.length() - 1] == ' ')
        result.pop_back();

    return result;
}

int main() {
    string text;
    cout << "Enter a string with extra spaces: ";
```

```
    getline(cin, text);

    string remove_space = removeSpaces(text);

    cout << "Cleaned string: \"" << remove_space << "\"" << endl;
    return 0;
}
```

Output:

Conclusion: In problem 3 we learn how to Removing Extra White Space from any Code.

Problem no 04:Check valid and invalid identifier

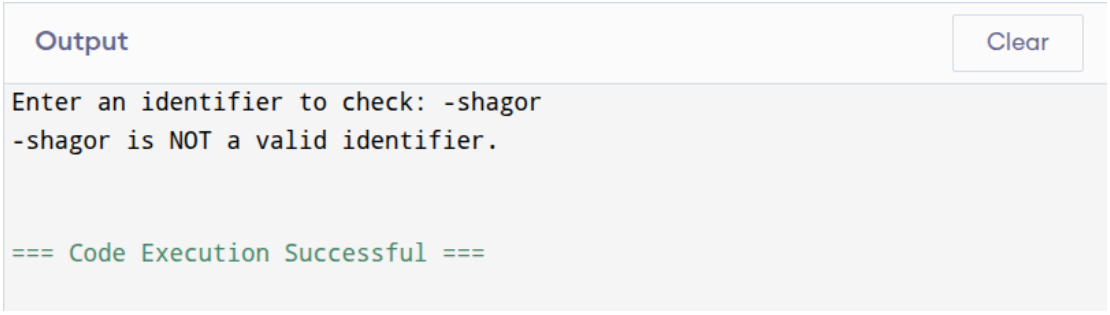
Software Requirements: Code Blocks

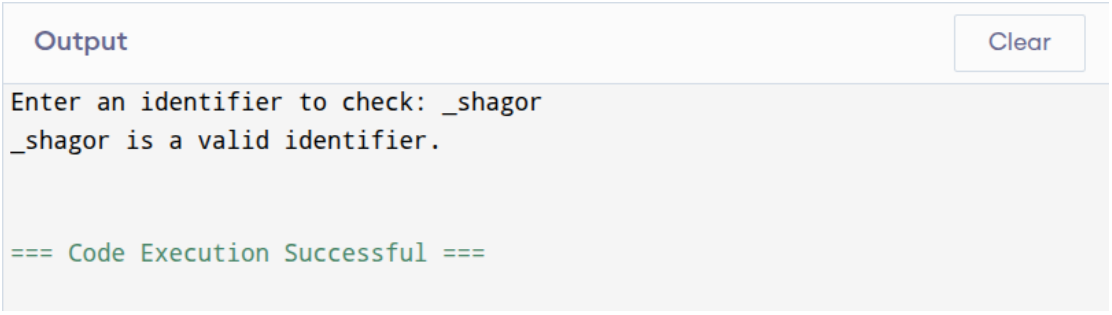
Source Code:

```
#include <iostream>
#include <cctype>

using namespace std;

bool isValidIdentifier(const string& id) {
    if (id.empty()) return false;

    if (!isalpha(id[0]) && id[0] != '_') return false;
```

```
    for (size_t i = 1; i < id.length(); ++i) {
        if (!isalnum(id[i]) && id[i] != '_') {
            return false;
        }
    }

    return true;
}

int main() {
    string identifier;
    cout << "Enter an identifier to check: ";
    cin >> identifier;

    if (isValidIdentifier(identifier)) {
        cout << identifier << " is a valid identifier." << endl;
    } else {
        cout << identifier << " is NOT a valid identifier." << endl;
    }

    return 0;
}
```

Output:





Conclusion: In problem 4 we learn how to Check valid and invalid identifiers.

Problem no 06: Check and count the total no of words, number and special character in the given input.

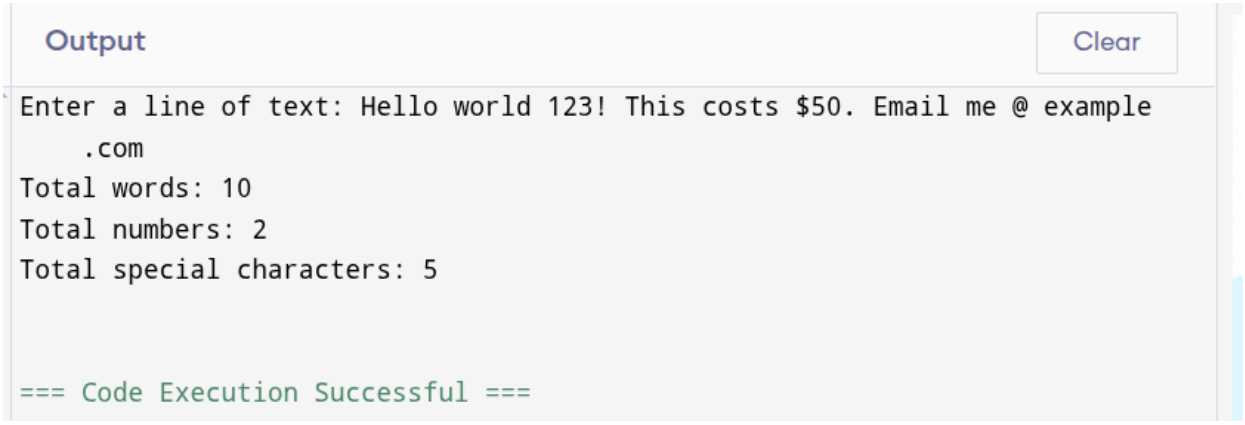
Software Requirements: Code Blocks

Source Code:

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

bool isNumber(const string& word) {
    for (char ch : word) {
        if (ch < '0' || ch > '9') return false;
    }
    return !word.empty();
}

int main() {
    string input;
    cout << "Enter a line of text: ";
    getline(cin, input);

    int wordCount = 0;
    int numberCount = 0;
    int specialCharCount = 0;

    string word = "";

    for (size_t i = 0; i < input.length(); ++i) {
        char ch = input[i];

        if ((ch >= 'a' && ch <= 'z') ||
            (ch >= 'A' && ch <= 'Z') ||
            (ch >= '0' && ch <= '9')) {
            word += ch;
        } else {

            if (!word.empty()) {
                ++wordCount;
```

```
            if (isNumber(word)) {
                ++numberCount;
            }
            word.clear();
        }

        if (ch != ' ') {
            ++specialCharCount;
        }
    }
}

if (!word.empty()) {
    ++wordCount;
    if (isNumber(word)) {
        ++numberCount;
    }
}

cout << "Total words: " << wordCount << endl;
cout << "Total numbers: " << numberCount << endl;
cout << "Total special characters: " << specialCharCount << endl;

return 0;
}
```

Output:



Conclusion: In problem 6 we learn how to Check and count the total number of words, number and special character in the given input.

Problem no 05: Regular expression.

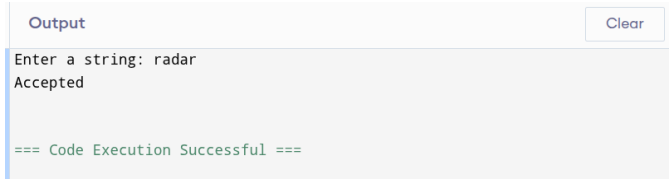
Problem no 5.1 : Start and ends with same symbols

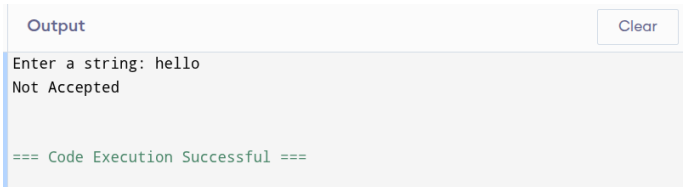
Software Requirements: Code Blocks

Source Code:

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

bool startsAndEndsSame(const string& str) {
    return !str.empty() && str.front() == str.back();
}

int main() {
    string input;

    cout << "Enter a string: ";
    getline(cin, input);

    if (startsAndEndsSame(input)) {
        cout << "Accepted" << endl;
    } else {
        cout << "Not Accepted" << endl;
    }

    return 0;
}
```

Output:





Conclusion: In problem 1 we learn how to define Start and End with the

same symbols.

Problem no 5.2 : Start and Ends with Different symbols

Software Requirements: Code Blocks

Source Code:

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

bool hasDifferentEnds(const string& str) {
    return !str.empty() && (str.front() != str.back());
}

int main() {
    string input;

    cout << "Enter a string: ";
    getline(cin, input);

    if (hasDifferentEnds(input)) {
        cout << "Accepted" << endl;
    } else {
        cout << "Not Accepted" << endl;
    }

    return 0;
}
```

Output:

Conclusion: In problem 2 we learn how to Define Start and End with Different
symbols.

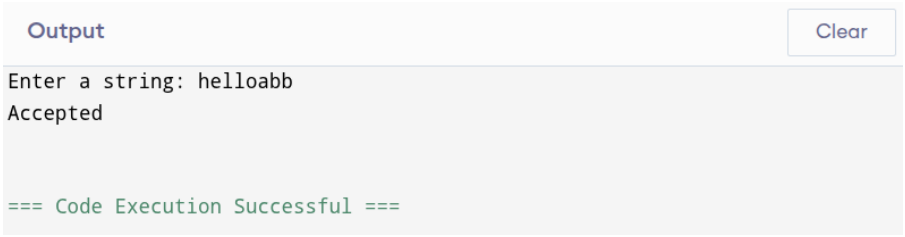
Problem no 5.3 : Ends with abb

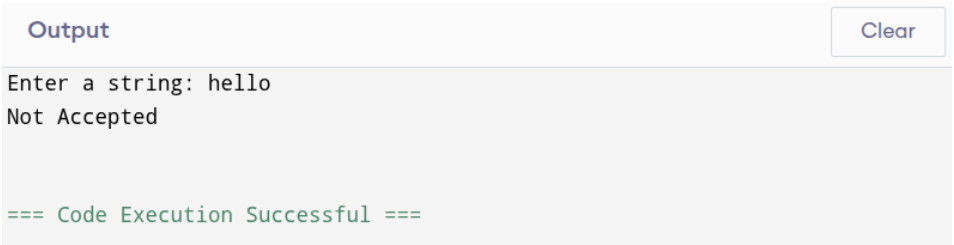
Software Requirements: Code Blocks

Source Code:

```
#include <iostream>
#include <string>
using namespace std;
int main() {
    string input;


    cout << "Enter a string: ";
    getline(cin, input);


    if (input.length() >= 3 && input.substr(input.length() - 3) == "abb") {
        cout << "Accepted" << endl;
    } else {
        cout << "Not Accepted" << endl;
    }

    return 0;
}
```

Output:





Conclusion: In problem 6 we learn how to Check string ends with abb symbol.