Muhammad Ahmad Raza

FA22-BCS-053-A

Compiler Lab

**Logical Operators:**

#include <iostream>

#include <regex>

#include <string>

using namespace std;

int main() {

regex re(R"([!=<>]=|&&|\|\||[><]=?)");

string str = "x == y && a > b || c <= d";

smatch match;

while (regex\_search(str, match, re)) {

cout << "match: " << match.str() << endl;

str = match.suffix();

}

return 0;

}

**Logical Expressions:**

#include <iostream>

#include <regex>

#include <string>

using namespace std;

int main() {

regex re(R"([a-zA-Z0-9]+(\s\*[\+\-\\*/\=\&\|\!<>\^\%\(\)]+)?[a-zA-Z0-9]+)");

string str = "x==y && a>b";

smatch match;

while (regex\_search(str, match, re)) {

cout << "match: " << match.str() << endl;

str = match.suffix();

}

return 0;

}

**Escape Sequence:**

#include <iostream>

#include <regex>

#include <string>

using namespace std;

int main() {

regex re(R"(\\[btnfr\"\'\\])");

string str = R"(x == y && a > b || c <= d \n \t \" \\ \n)";

smatch match;

while (regex\_search(str, match, re)) {

cout << "match: " << match.str() << endl;

str = match.suffix();

}

return 0;

}

**Everything in 1 file:**#include <iostream>

#include <regex>

#include <string>

using namespace std;

int main()

{

    // Integer

    regex re("[0-9]+");

    string str = "2+4";

    smatch match;

    while (regex\_search(str, match, re)) {

        cout << "Integer match: " << match.str() << endl;

        str = match.suffix();

    }

    // String identifier

    re = regex("[A-Za-z\_][0-9A-Za-z\_]\*");

    str = "ab+a7+\_3";

    while (regex\_search(str, match, re)) {

        cout << "String Identifier match: " << match.str() << endl;

        str = match.suffix();

    }

    // Operator Matching

    re = regex("[+\\-\*/%]");

    str = "ab+a7+\_3";

    while (regex\_search(str, match, re)) {

        cout << "Operator Matching match: " << match.str() << endl;

        str = match.suffix();

    }

    // Floating-Point Numbers

    re = regex("[0-9]+\\.[0-9]+");

    str = "4.3+8.1";

    while (regex\_search(str, match, re)) {

        cout << "Floating Point numbers match: " << match.str() << endl;

        str = match.suffix();

    }

    // Exponential

    re = regex("[0-9]+e\\^[+\\-]?[0-9]+");

    str = "24e^5";

    while (regex\_search(str, match, re)) {

        cout << "Exponential match: " << match.str() << endl;

        str = match.suffix();

    }

    // Keywords match

    re = regex("(if)|(cout)|(endl)");

    str = "if(count<<endl)";

    while (regex\_search(str, match, re)) {

        cout << "Keywords match: " << match.str() << endl;

        str = match.suffix();

    }

    // Logic operators

    re = regex(R"([!=<>]=|&&|\|\||[><]=?)");

    str = "x == y && a > b || c <= d";

    while (regex\_search(str, match, re)) {

        cout << "Logical Operators match: " << match.str() << endl;

        str = match.suffix();

    }

    // Logical Expressions

    re = regex(R"([a-zA-Z0-9]+(\s\*[\+\-\\*/\=\&\|\!<>\^\%\(\)]+)?[a-zA-Z0-9]+)");

    str = "x==y && a>b";

    while (regex\_search(str, match, re)) {

        cout << "Logical Expressions match: " << match.str() << endl;

        str = match.suffix();

    }

    // Escape Sequence

    re = regex(R"(\\[btnfr\"\'\\])");

    str = R"(x == y && a > b || c <= d \n \t \" \\ \n)";

    while (regex\_search(str, match, re)) {

        cout << "Escape Sequence match: " << match.str() << endl;

        str = match.suffix();

    }

    return 0;

}

Output:  
