



GHANA COMMUNICATION TECHNOLOGY UNIVERSITY

INSTITUTE OF CONTINUING AND DISTANCE EDUCATION (ICDE)

COURSE CODE	CICS 112
COURSE TITLE	Programming with C++
DATE	22 nd August, 2025
GROUP NAME	GROUP 13

INDEX NUMBER	NAME
2425140073	Francklina Delali Mensah
2425140016	RUNCIE FRIMPONG
2425140023	AGBENYO DELATOR ROGERS
2425140099	Darglo Isaac Mawuli
2425140125	Emmanuel Koomson

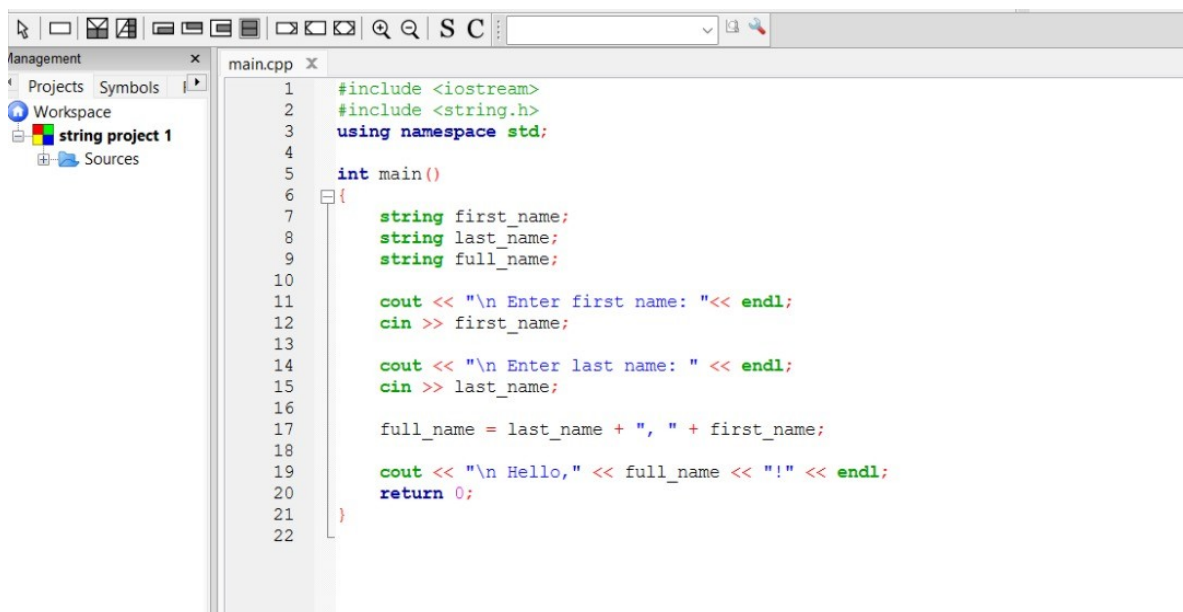
Question 1

Write a C++ program that accepts a user's first name and last name as separate string inputs, then combines them into a full name and displays it in the format: "Hello, [Last Name], [First Name]!"

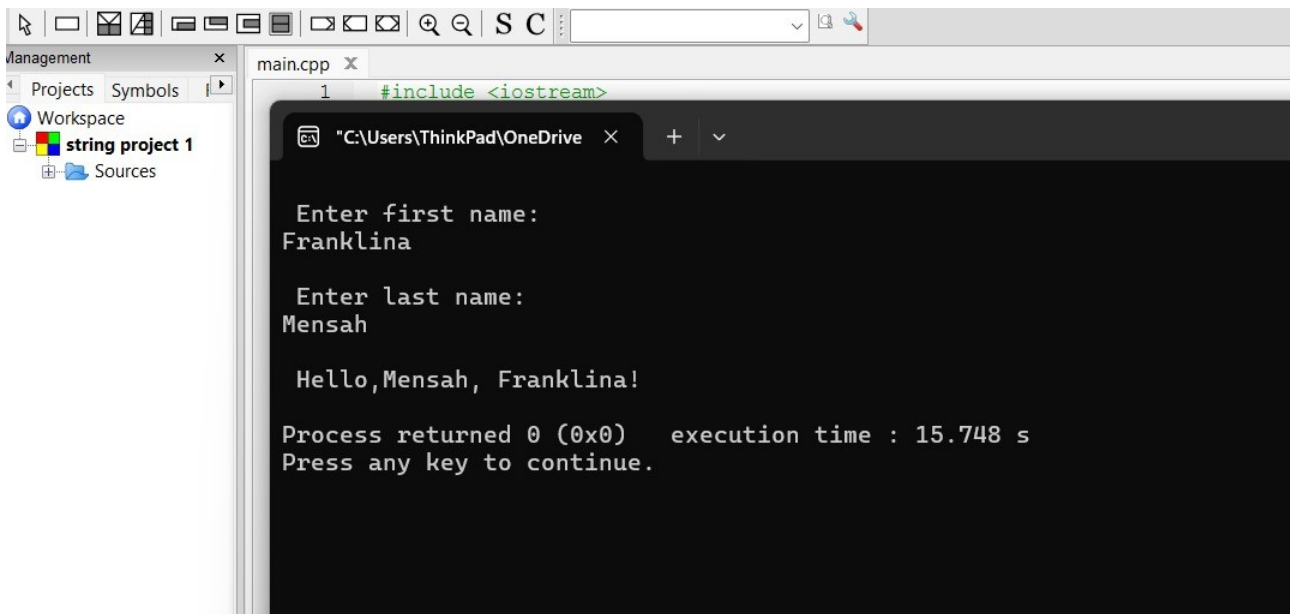
The Solution

```
#include <iostream>
#include <string.h>
using namespace std;
int main()
{
    string first_name;
    string last_name;
    string full_name;
    cout << "\n Enter first name: " << endl;
    cin >> first_name;
    cout << "\n Enter last name: " << endl;
    cin >> last_name;
    full_name = last_name + ", " + first_name;
    cout << "\n Hello," << full_name << "!" << endl;
    return 0;
}
```

Screenshot Of The Source Code



Screenshot of the Output



Question 2

Create a C++ program that prompts the user to enter a password. The program should check if the password contains at least 8 characters and includes the word "GCTU" as part of it. Display a message indicating whether the password is valid or not.

The Solution

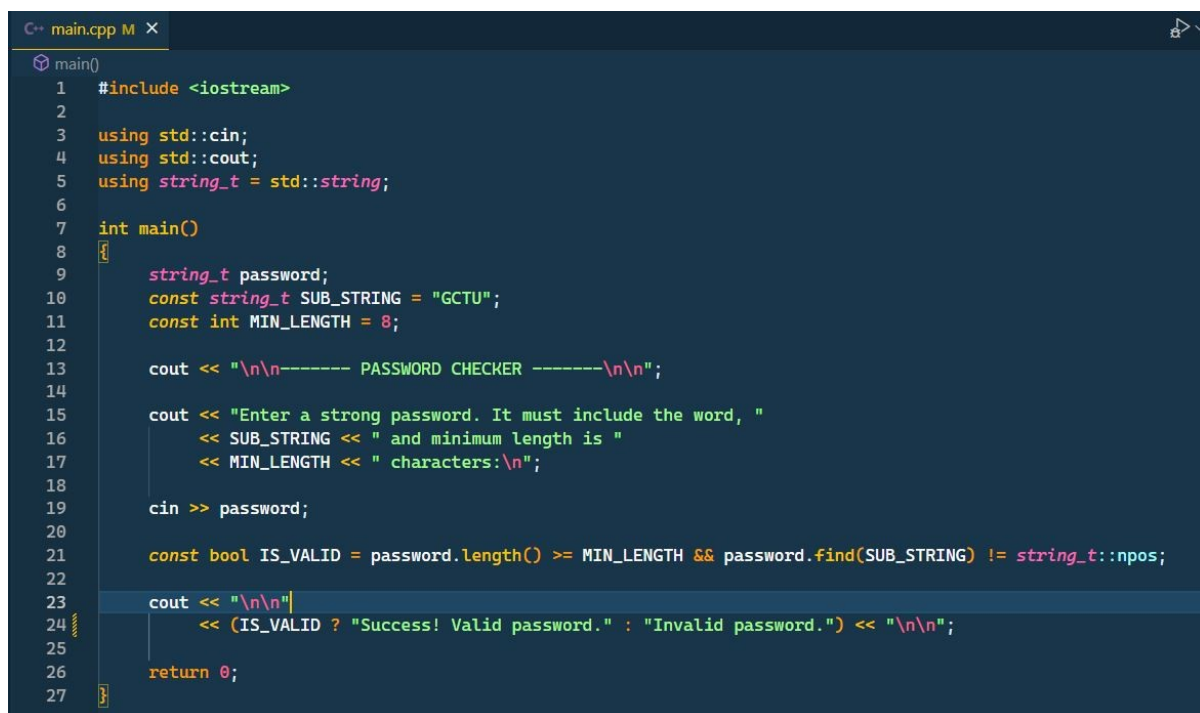
```
#include <iostream>
using std::cin;
using std::cout;
using string_t = std::string;
```

```

int main()
{
    string_t password;
    const string_t SUB_STRING = "GCTU";
    const int MIN_LENGTH = 8;
    cout << "\n\n----- PASSWORD CHECKER ----- \n\n";
    cout << "Enter a strong password. It must include the word, "
        << SUB_STRING << " and minimum length is "
        << MIN_LENGTH << " characters:\n";
    cin >> password;
    const bool IS_VALID = password.length() >= MIN_LENGTH &&
password.find(SUB_STRING) != string_t::npos;
    cout << "\n\n" << (IS_VALID ? "Success! Valid password." : "Invalid password.") << "\n\n";
    return 0;
}

```

Screenshot Of The Source Code



```

C++ main.cpp M x
main()
1  #include <iostream>
2
3  using std::cin;
4  using std::cout;
5  using string_t = std::string;
6
7  int main()
8  {
9      string_t password;
10     const string_t SUB_STRING = "GCTU";
11     const int MIN_LENGTH = 8;
12
13     cout << "\n\n----- PASSWORD CHECKER ----- \n\n";
14
15     cout << "Enter a strong password. It must include the word, "
16         << SUB_STRING << " and minimum length is "
17         << MIN_LENGTH << " characters:\n";
18
19     cin >> password;
20
21     const bool IS_VALID = password.length() >= MIN_LENGTH && password.find(SUB_STRING) != string_t::npos;
22
23     cout << "\n\n"
24         << (IS_VALID ? "Success! Valid password." : "Invalid password.") << "\n\n";
25
26     return 0;
27 }

```

Screenshot Of The Output

```
delanyo@thedelanyo MINGW64 /d/dev/gctu_projects/second_sem/cpp/password_checker (main)
• $ ./main

----- PASSWORD CHECKER -----

Enter a strong password. It must include the word, GCTU and minimum length is 8 characters:
pass123#etebkj%ghh

Invalid password. Must include GCTU and a min of 8 chars.

delanyo@thedelanyo MINGW64 /d/dev/gctu_projects/second_sem/cpp/password_checker (main)
• $ ./main

----- PASSWORD CHECKER -----

Enter a strong password. It must include the word, GCTU and minimum length is 8 characters:
GCTU123

Invalid password. Must include GCTU and a min of 8 chars.

delanyo@thedelanyo MINGW64 /d/dev/gctu_projects/second_sem/cpp/password_checker (main)
• $ ./main

----- PASSWORD CHECKER -----

Enter a strong password. It must include the word, GCTU and minimum length is 8 characters:
pass#1233@GCTU

Success! Valid password.

delanyo@thedelanyo MINGW64 /d/dev/gctu_projects/second_sem/cpp/password_checker (main)
○ $ █
```

Question 3

Write a C++ program that accepts a sentence from the user and counts how many words are in the sentence. Assume words are separated by spaces.

The Solution

```
#include <iostream>
#include<string>
using namespace std;
int main()
{string sentence;
int wordcount =0;
cout << "Enter a sentence:";
getline(cin,sentence); // Accepts entire sentence including spaces
bool inWord = false;
for (char c:sentence){
if (c!=' ' && !inWord) {
//Start of a new word
inWord = true;
wordcount++;
}
else if (c==' '){
//End of a word
inWord = false;
}
}
cout <<"Number of words in the sentence:" <<
wordcount <<endl;
return 0;
}
```

Screenshot Of The Source Code

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

int main()
{string sentence;
int wordcount =0;

cout << "Enter a sentence:";
getline(cin,sentence); // Accepts entire sentence including spaces

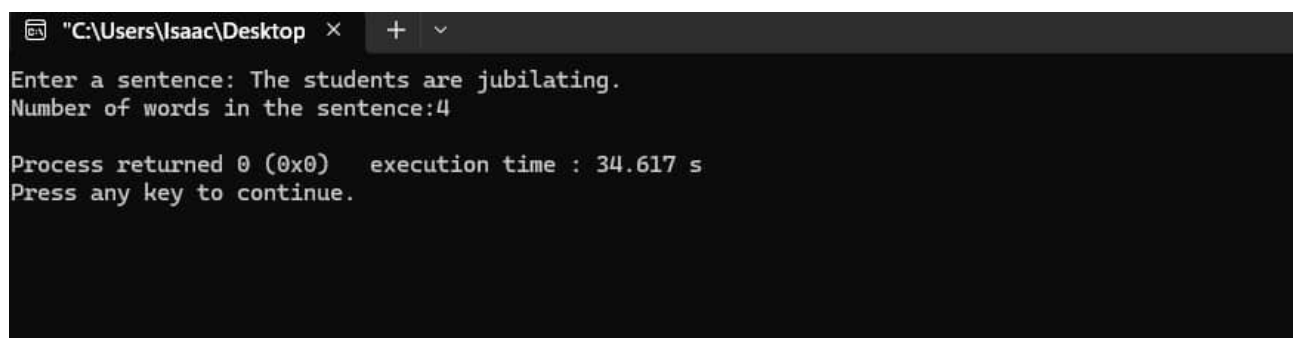
bool inWord = false;

for (char c:sentence){
    if (c!=' ' && !inWord) {
        //Start of a new word
        inWord = true;
        wordcount++;
    }
    else if (c==' '){
        //End of a word
        inWord = false;
    }
}

cout <<"Number of words in the sentence:" <<
wordcount <<endl;

return 0;
}
```

The Output



```
"C:\Users\Isaac\Desktop" x + v
Enter a sentence: The students are jubilating.
Number of words in the sentence:4

Process returned 0 (0x0)   execution time : 34.617 s
Press any key to continue.
```

Question 4

As software engineer, write a C++ program to solve the quadratic equation $y = ax^2 + bx + c$. using constants for the coefficients a, b, and c.

The Solution

```
#include <iostream>
#include <cmath>
using std::cin;
using std::cout;
using std::sqrt;
int main()
{
    cout << "\n\n-----  QUADRATIC EQUATION SOLVER  ----- \n\n";
    double a, b, c;
    cout << "Enter coefficients a, b, c: ";
    cin >> a >> b >> c;
    cout << "\n\n";
    if (a == 0)
    {
        cout << "Not a quadratic equation (a must not be zero)";
        return 0;
    }
    double discriminant = b * b - 4 * a * c;
    if (discriminant > 0)
    {
        // two real roots
        double root1 = (-b + sqrt(discriminant)) / (2 * a);
        double root2 = (-b - sqrt(discriminant)) / (2 * a);

        cout << "Two real roots: " << root1 << " and " << root2;
    }
    else if (discriminant == 0)
    {
        // one real root
        double root = -b / (2 * a);

        cout << "One real root: " << root;
    }
    else
    {
        // complex roots
        double realPart = -b / (2 * a);
        double imaginaryPart = sqrt(-discriminant) / (2 * a);

        cout << "Complex roots: "
            << realPart << " + " << imaginaryPart << "i and "
```



```

        << realPart << " - " << imaginaryPart << "i";
    }
    cout << "\n\n";
    return 0;
}

```

Screenshot Of The Source Code



```

1  #include <iostream>
2  #include <cmath>
3
4  using std::cin;
5  using std::cout;
6  using std::sqrt;
7
8  int main()
9  {
10
11     cout << "\n\n-----  QUADRATIC EQUATION SOLVER  ----- \n\n";
12
13     double a, b, c;
14     cout << "Enter coefficients a, b, c: ";
15     cin >> a >> b >> c;
16
17     cout << "\n\n";
18
19     if (a == 0)
20     {
21         cout << "Not a quadratic equation (a must not be zero)";
22         return 0;
23     }
24
25     double discriminant = b * b - 4 * a * c;
26
27     if (discriminant > 0)
28     {
29         // two real roots
30         double root1 = (-b + sqrt(discriminant)) / (2 * a);
31         double root2 = (-b - sqrt(discriminant)) / (2 * a);
32
33         cout << "Two real roots: " << root1 << " and " << root2;
34     }
35     else if (discriminant == 0)
36     {
37         // one real root
38         double root = -b / (2 * a);
39
40         cout << "One real root: " << root;
41     }
42     else
43     {
44         // complex roots
45         double realPart = -b / (2 * a);
46         double imaginaryPart = sqrt(-discriminant) / (2 * a);
47
48         cout << "Complex roots: "
49             << realPart << " + " << imaginaryPart << "i and "
50             << realPart << " - " << imaginaryPart << "i";
51     }
52
53     cout << "\n\n";
54
55     return 0;
56 }
57

```

Screenshot Of The Output

```
deLanyo@thedeLanyo MINGW64 /d/dev/cpp/gctu/quadratic_equation
● $ ./main

-----    QUADRATIC EQUATION SOLVER    -----

Enter coefficients a, b, c: 2 -5 3

Two real roots: 1.5 and 1

deLanyo@thedeLanyo MINGW64 /d/dev/cpp/gctu/quadratic_equation
○ $ █
```

Question 5

Write C++ Program to build a simple calculator using switch Statement.

The Solution

```
#include <iostream>

int main()
{
    double first, second, result;
    char operate;

    std::cout << "Enter your first number: ";
    std::cin >> first;
    std::cout << "Enter operator (+, -, *, /): ";
    std::cin >> operate;

    std::cout << "Enter your second number: ";
    std::cin >> second;

    switch (operate)
    {
        case '+':
            result = first + second;
            break;

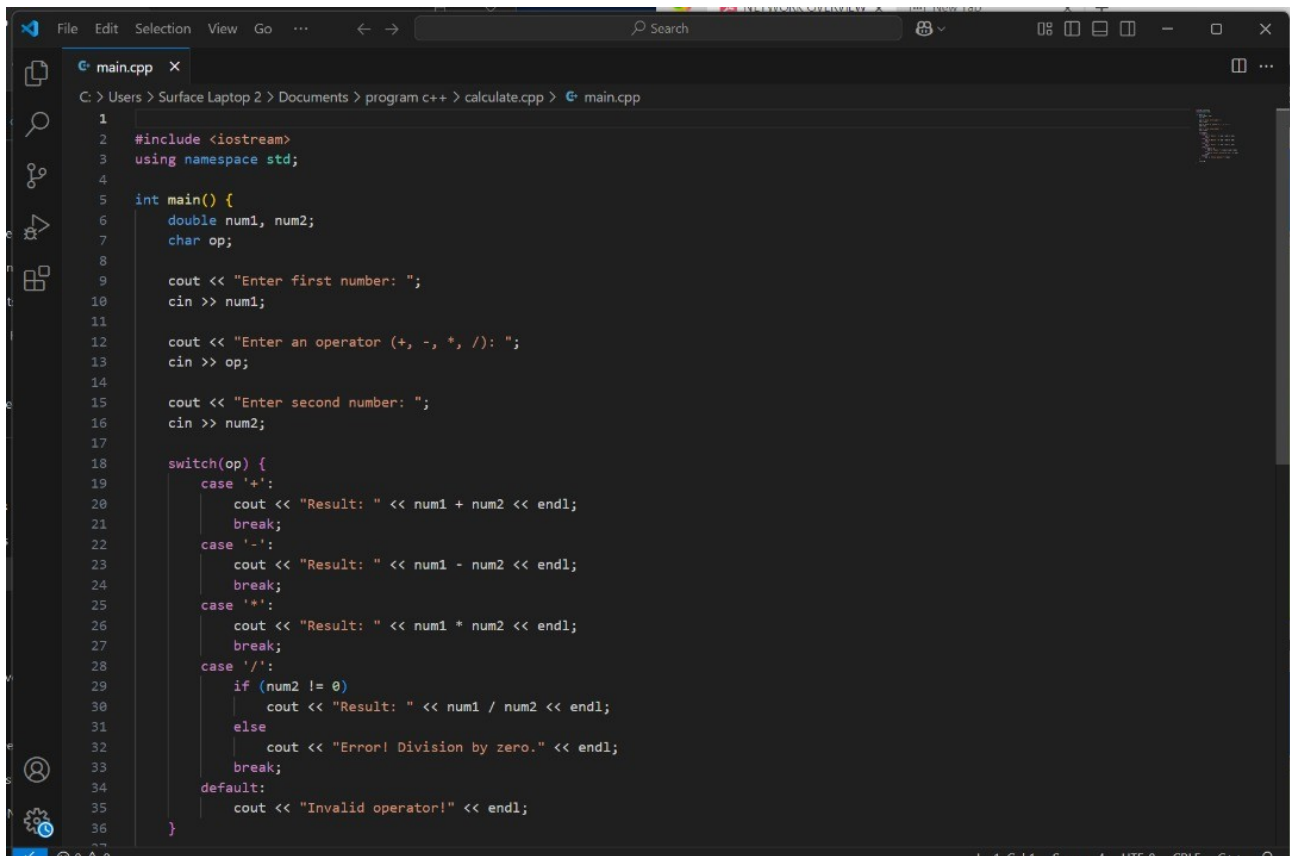
        case '-':
            result = first - second;
            break;

        case '*':
            result = first * second;
            break;

        case '/':
            result = first / second;
            return 1;

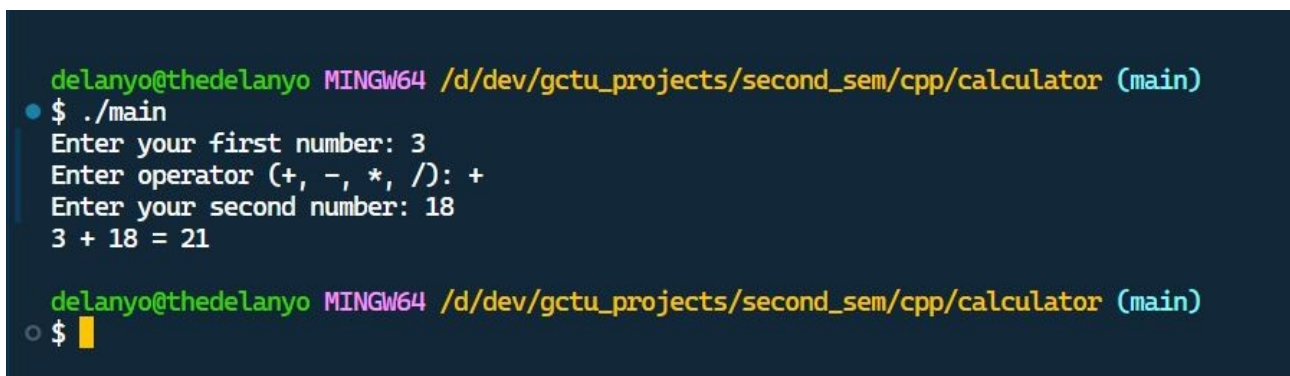
        default:
            std::cout << "Invalid operator";
    }
    std::cout << first << " " << operate << " " << second << " = " << result << std::endl;
    return 0;
}
```

Screenshot Of The Source Code



```
1
2 #include <iostream>
3 using namespace std;
4
5 int main() {
6     double num1, num2;
7     char op;
8
9     cout << "Enter first number: ";
10    cin >> num1;
11
12    cout << "Enter an operator (+, -, *, /): ";
13    cin >> op;
14
15    cout << "Enter second number: ";
16    cin >> num2;
17
18    switch(op) {
19        case '+':
20            cout << "Result: " << num1 + num2 << endl;
21            break;
22        case '-':
23            cout << "Result: " << num1 - num2 << endl;
24            break;
25        case '*':
26            cout << "Result: " << num1 * num2 << endl;
27            break;
28        case '/':
29            if (num2 != 0)
30                cout << "Result: " << num1 / num2 << endl;
31            else
32                cout << "Error! Division by zero." << endl;
33            break;
34        default:
35            cout << "Invalid operator!" << endl;
36    }
```

Screenshot Of The Output



```
delanyo@thedelanyo MINGW64 /d/dev/gctu_projects/second_sem/cpp/calculator (main)
$ ./main
Enter your first number: 3
Enter operator (+, -, *, /): +
Enter your second number: 18
3 + 18 = 21

delanyo@thedelanyo MINGW64 /d/dev/gctu_projects/second_sem/cpp/calculator (main)
$
```