

# Simple Calculator

ეს პროგრამა წარმოადგენს მარტივ კალკულატორი, რომლის მეშვეობით მომხმარებელს შეუძლია გამოთვალოს ორი რიცხვის ჯამი, სხვაობა, ნამრავლი ან განაყოფი.

პროგრამის ჩართვისას მომხმარებელი ხედავს შემდეგ შეტყობინებას:

```
-= [Simple Calculator] -=  
Enter an operator (+, -, *, /) to start or 'q' to quit: █
```

მას შეუძლია 'q' სიმბოლო შეიტანოს პროგრამის დასასრულებლად:

```
Enter an operator (+, -, *, /) to start or 'q' to quit: q  
Come back soon <3
```

ან ოთხი ოპერატორიდან ერთ-ერთი:

```
Enter an operator (+, -, *, /) to start or 'q' to quit: +  
x + y = ?
```

არავალიდური ინფუთის შემთხვევაში პროგრამა გამოიტანს შესაბამის შეტყობინებას, რის შემდეგ მომხმარებელი შეძლებს თავიდან შეიტანოს მონაცემები:

```
Enter an operator (+, -, *, /) to start or 'q' to quit: abc  
>>YOU MUST ENTER AN OPERATOR<<  
Try again  
Enter an operator (+, -, *, /) to start or 'q' to quit: █
```

თუ მომხმარებელი შეიტანს რამდენიმე ოპერატორს ერთდროულად, პროგრამა წაიკითხავს პირველს:

```
Enter an operator (+, -, *, /) to start or 'q' to quit: +/*  
x + y = ?
```

შემდეგ მომხმარებელმა უნდა შეიტანოს ორი ოპერანდი:

```
Enter first number: 4
4 + y = ?

Enter second number: 6
4 + 6 = 10
```

არავალიდური ინფუთის შემთხვევაში პროგრამა გამოიტანს შესაბამის შეტყობინებას, რის შემდეგ მომხმარებელი შეძლებს თავიდან შეიტანოს მონაცემები:

<pre>Enter first number: abc  &gt;&gt;YOU MUST ENTER A NUMBER&lt;&lt; Try again  Enter first number: </pre>	<pre>Enter second number: efg  &gt;&gt;YOU MUST ENTER A NUMBER&lt;&lt; Try again  Enter second number: </pre>
---	---

ასევე იქნება გამოტანილი შეტყობინება, თუ მოხდება ნულზე გაყოფა

```
Enter an operator (+, -, *, /) to start or 'q' to quit: /
x / y = ?

Enter first number: 2
2 / y = ?

Enter second number: 0

>>YOU CAN'T DIVIDE BY ZERO<<
Try again
```

პასუხის გამოტანის შემდეგ პროგრამა თავიდან ჩაირთვება:

```
4 + 5 = 9

Enter an operator (+, -, *, /) to start or 'q' to quit: 
```

## კოდი

```
// -=[Simple Calculator]=-

#include <iostream>
#include <limits>
using namespace std;

// inputOper() function receives one of four operators and starts program
// or receives 'q' and quits program
void inputOper(char *op)
{
    while (true)
    {
        cout << "Enter an operator (+, -, *, /) to start or 'q' to quit: ";
        cin >> *op;
        cin.ignore(numeric_limits<streamsize>::max(), '\n');
        // ignores the rest of characters
        if (*op == 'q')
        {
            cout << "\nCome back soon <3\n";
            exit(0); // program ends
        }
        else if (*op != '+' && *op != '-' && *op != '*' && *op != '/')
            // checks if character is an operator
        {
            cerr << "\n>>YOU MUST ENTER AN OPERATOR<<\nTry again\n\n";
            continue; // lets user input operator again if input failed
        }
        cout << "x " << *op << " y = ?\n\n";
        break;
    }
}

// inputNums() function receives two numbers one by one
void inputNums(double *x, double *y, const char *op)
{
    while (true)
    {
        cout << "Enter first number: ";
        cin >> *x;
        if (cin.fail()) // if user didn't input a number
        {
            cin.clear(); // clears input
        }
    }
}
```

```

        cin.ignore(numeric_limits<streamsize>::max(), '\n');
        // ignores the rest of characters
        cerr << "\n>>YOU MUST ENTER A NUMBER<<\nTry again\n\n";
        continue; // lets user input number again if input failed
    }
    else
    {
        cout << *x << " " << *op << " y = ?\n\n";
        break;
    }
}

while (true)
{
    cout << "Enter second number: ";
    cin >> *y;
    if (cin.fail()) // if user didn't input a number
    {
        cin.clear(); // clears input
        cin.ignore(numeric_limits<streamsize>::max(), '\n');
        // ignores the rest of characters
        cerr << "\n>>YOU MUST ENTER A NUMBER<<\nTry again\n\n";
        continue;
    }
    else
    {
        if (*op == '/' && *y == 0) // checks if there is division by zero
        {
            cerr << "\n>>YOU CAN'T DIVIDE BY ZERO<<\nTry again\n\n";
            continue; // lets user input number again if input failed
        }
        else
        {
            break;
        }
    }
}

// calc() function calculates and prints result according to operator
void calc(double *x, double *y, char *op)
{
    switch (*op)
    {
        case '+':
            cout << *x << " + " << *y << " = " << *x + *y << "\n";
            break;
        case '-':

```

```

        cout << *x << " - " << *y << " = " << *x - *y << "\n";
        break;
    case '*':
        cout << *x << " * " << *y << " = " << *x * *y << "\n";
        break;
    case '/':
        cout << *x << " / " << *y << " = " << *x / *y << "\n";
        break;
    }
    cout << endl;
}

int main()
{
    char *oper = new (nothrow) char;
    double *num1 = new (nothrow) double;
    double *num2 = new (nothrow) double;

    cout << "=[Simple Calculator]=-\n\n";
    while (true)
    {
        inputOper(oper);
        inputNums(num1, num2, oper);
        calc(num1, num2, oper);
    }
}

```