

Forum:[Hack Forums](#)

Thread: [\[Mini-Tutorial\] An easy Calculator \[Showcase\]](#)

Niely - March 22, 2014, 5:04 am

Hello

I'm still a beginner in C++ but with the knowledge I already have I succeeded to make a MINI-calculator in C++.

I thought it'd be cool to share it with you and maybe even post a mini-tutorial about it. :)

Like I said, I'm still a beginner so correct me if I'm wrong!

Things we will use

Cout<<"String"<<endl; To display/echo things on your screen.

Else/if statement To do something if user input is True/False

Cin >> Example; To make an user input.

Int var1, var2, var3; to say what variables we use.

Goto Label

Beginning To begin with your calculator you should have:

Code #include <iostream>

Using namespace std;

Int main() {

}

EVERY code for the calculator should be between the brackets of main.

Cout

We use cout to display things on our screen, mostly strings (strings are everything between " " quotes), but also numbers or variable.

Examples:

Cout<<"Hello there"<<endl; String

Cout<<5<<endl; Number with VALUE

But we can also display variables with it:

Code int myVariable = 123

cout<<myVariable<<endl; This will display 123.

Cout<<"6"<<endl; Is NOT the same as cout<<6<<endl; because the second has a value, in this case 6. The first is just a string without value. :)

If you don't use <<endl; but just: cout<<"Hello";

This will display something but not jump to the next line.

Conclusion:

With cout we can display things

With cout we can display the value/input of a variable

With <<endl; we let our cursor jump to the next line.

We can display strings and values

Variables

Variables are actually little words/characters we give a value of meaning.

Every variable should be mentioned with int.

With int we say we use variables and how they call.

Example:

```
int variable1, variable2, var, total, myVariable;
```

You can set every variable on one line with int like we did above... But you can also separate them.

Example:

```
Code Int x = 7;
```

```
Cout<<x<<endl;
```

```
Int y = 4;
```

```
Cout<<y<<endl;
```

This will display:

7

4

Cin

With cin we can make an user input.

Example:

```
Cin >> myVariable;
```

Keep this in mind for the if/else statement! :)

If/else statement

The if/else statement for C++ looks a lot like the one in JavaScript

```
Code If (Statement) {
```

```
Do something
```

```
}
```

```
Else if (statement) {
```

```
Do something
```

```
}
```

```
Else {
```

```
Do something if none of above works/is true.
```

```
}
```

We already know we can give variables a value, let's try it!

```
Code Int x = 8;
```

```
If (x==8) {
```

```
Cout<<"True"<<endl;
```

```
}
```

```
Else {
```

```
Cout<<"Error"<<endl;
```

```
}
```

I said you to keep the cin option in mind, we can use it like this in the if/else statement:

```
Code Int x;
```

```
Cin >> x;
```

```
If (x==8) {
```

```
Cout<<"True"<<endl;
```

```

}
Else if (x==58) {
Cout<<"Hey"<<endl;
}
Else {
Cout<<"Error"<<endl;
}
}
If user input (cin) is equal to eight, display True, or if x is equal to fifty-eight display Hey else
display Error."
You see? :)

```

You can have as much else if as you want!

Labels and GoTo

Another useful piece of code is GoTo.

It looks like this:

Code myLabel:

LOT OF CODE

GoTo myLabel

Where myLabel is the label, a label should always end with ":". With GoTo you can jump to that label and don't have to re-write the whole piece of code each time, but you can just 'jump' to it. :)

The Calculator Engine

Now the engine, the engine is very simple and based on the if/else statement.

The total engine to sum, looks like this:

```

Code    cout<<"Enter numbers to sum beneath eachother"<<endl;
        cin >> sum1;
        cout<<"+"<<endl;
        cin >> sum2;
        cout<<"="<<endl;
        sum = sum1 + sum2;
        cout<<" "<<endl;
        cout<<sum<<endl;
        cout<<" "<<endl;
        cout<<"_____"<<endl;
        goto Start;

```

Cin is the user input and has variable: sum1 for the first number; then we have another cin for the second number.

And the sum (Variable Sum) is equal to: Sum1 (number1) + sum2 (number2)

And then we cout sum (the total).

All other couts are to make your output look nice, you can style it like you want.

And at the end of all, we go back to the beginning, so the user can calculate other things again without re-starting it.

If you know the engine for subtracting is like this:

```

Code    cout<<"Enter numbers to subtract beneath eachother"<<endl;
        cin >> subs1;
        cout<<"-"<<endl;
        cin >> subs2;
        cout<<"="<<endl;

```

```

subtraction = subs1 - subs2;
cout<<" "<<endl;
cout<<subtraction<<endl;
cout<<" "<<endl;
cout<<"_____"<<endl;
goto Start;

```

And to multiply like this:

```

Code cout<<"Enter numbers to multiply beneath eachother"<<endl;
cin >> mul1;
cout<<"x"<<endl;
cin >> mul2;
cout<<"="<<endl;
mul = mul1 * mul2;
cout<<" "<<endl;
cout<<mul<<endl;
cout<<" "<<endl;
cout<<"_____"<<endl;
goto Start;

```

How will it looks to divide if you know it can be done with "/"?

Sum: +
 Subtract: -
 Multiply: *
 Divide: /

Make sure to put all your variables in an int!

Choose what you want to do

But we also have to let the user decide what they want to do, we do that with the else/if statement and cin.

```

Code Cout<<"Press 1 to do this"<<endl;
Cout<<"Press 2 to do another thing"<<endl;
Cin >> Option;

```

```

If (option==1) {
Cout<<"This"<<endl;
}
Else if (option==2) {
Cout<<"Another thing"<<endl;
}
Else {
Cout<<"ERROR!!!"<<endl;
}

```

With this knowledge you can make your own calculator! :D

Total Source Code:

```

Code #include <iostream>
using namespace std;

```

```

int main() {
int option, sum1, sum2, sum, subs1, subs2, subtraction, mul1, mul2, mul, div1, div2, div;
cout<<"~Calculator~"<<endl;

```

```

cout<<" "<<endl;
cout<<"Calculate Options:"<<endl;
cout<<"Press 1 to sum two numbers"<<endl;
cout<<"Press 2 to subtract two numbers"<<endl;
cout<<"Press 3 to multiply two numbers"<<endl;
cout<<"Press 4 to divide two numbers"<<endl;
cout<<" "<<endl;
cout<<"Other options:"<<endl;
cout<<"Enter 5 to view who made this"<<endl;
cout<<"Enter 6 to read about this calculator"<<endl;
Start:
cin >> option;
if (option==1) {
    cout<<"Enter numbers to sum beneath eachother"<<endl;
    cin >> sum1;
    cout<<"+"<<endl;
    cin >> sum2;
    cout<<"="<<endl;
    sum = sum1 + sum2;
    cout<<" "<<endl;
    cout<<sum<<endl;
    cout<<" "<<endl;
    cout<<"_____"<<endl;
    goto Start;
}
else if (option==2) {
    cout<<"Enter numbers to subtract beneath eachother"<<endl;
    cin >> subs1;
    cout<<"-"<<endl;
    cin >> subs2;
    cout<<"="<<endl;
    subtraction = subs1 - subs2;
    cout<<" "<<endl;
    cout<<subtraction<<endl;
    cout<<" "<<endl;
    cout<<"_____"<<endl;
    goto Start;
}
else if (option==3) {
    cout<<"Enter numbers to multiply beneath eachother"<<endl;
    cin >> mul1;
    cout<<"x"<<endl;
    cin >> mul2;
    cout<<"="<<endl;
    mul = mul1 * mul2;
    cout<<" "<<endl;
    cout<<mul<<endl;
    cout<<" "<<endl;
    cout<<"_____"<<endl;
    goto Start;
}
else if (option==4) {
    cout<<"Enter numbers to divide beneath eachother"<<endl;

```

```

cin >> div1;
cout<<"."<<endl;
cin >> div2;
cout<<"="<<endl;
div = div1 / div2;
cout<<" "<<endl;
cout<<div<<endl;
cout<<" "<<endl;
cout<<"_____"<<endl;
goto Start;
}
else if (option==5) {
    cout<<"About me"<<endl;
    cout<<"This calculator is made by Niely"<<endl;
    cout<<"Webmaster of www.iCode4U.tk"<<endl;
    cout<<" "<<endl;
    cout<<"_____"<<endl;
    goto Start;
}
else if (option==6) {
    cout<<"About Calculator"<<endl;
    cout<<"This calculator is made for a tutorial for Hackforums.net"<<endl;
    cout<<"This calculator can calculate: +,-,x,:. Good Luck! :D"<<endl;
    cout<<" "<<endl;
    cout<<"_____"<<endl;
    goto Start;
}
else {
    cout<<"Option not found!"<<endl;
    goto Start;
}
}
}

```

Sorry for my bad English/Broken explanation, just wanted to share! :D

Thanks for reading,
Niely

g4143 - March 22, 2014, 5:06 am

Could you explain why you need goto's in your program?

This is terrible style.

Code if (option==1)

```
{...}
```

It should be.

Code if (option == 1)

```
{...}
```

Why all the variables? You should be able to get away with three int variables.

Code int option, sum1, sum2, sum, subs1, subs2, subtraction, mul1, mul2, mul, div1, div2,

div;//why so many? Here's a shell that you can start with. Please note that I haven't worked with C++'s IO in a while so it may be buggy.

Code #include <iostream>

```

struct values
{
    int v1;
    int v2;
};

struct values get_values()
{
    struct values ans = {0, 0};

    std::cout << "Enter two numbers seperated by a space->";
    std::cin >> ans.v1 >> ans.v2;

    return ans;
}

void pause()
{
    std::cout << "Please hit any key to continue..." << std::endl;
    std::cin.ignore();
    std::cin.get();
}

int main()
{
    int option;
    bool looping = true;

    while ( looping )
    {
        std::cout << "The simple calculator" << std::endl;
        std::cout << "Enter 0 to exit" << std::endl;
        std::cout << "Enter 1 for addition" << std::endl;
        std::cout << "Enter 2 for subtraction" << std::endl;

        std::endl(std::cout);

        std::cout << "Enter choice->";
        std::cin >> option;

        switch ( option )
        {
            case 0:
            {
                looping = false;
                break;
            }
            case 1:
            {
                struct values v = get_values();
                std::cout << v.v1 << " + " << v.v2 << " = " << v.v1 + v.v2 << std::endl;
                std::endl(std::cout);
                pause();
            }
        }
    }
}

```

```

break;
}
case 2:
{
struct values v = get_values();
std::cout << v.v1 << " - " << v.v2 << " = " << v.v1 - v.v2 << std::endl;
std::endl(std::cout);
pause();

break;
}
default:
{
std::cerr << "Invalid choice->" << option << std::endl;
pause();
}
}
}
}
}

```

Niely - March 22, 2014, 6:19 am

^Thanks for your help! :)

For goto and label:

Quote:you can jump to that label and don't have to re-write the whole piece of code each time, but you can just 'jump' to it. :)

But that's a really advanced/hard to understand code I think, it won't be an easy calculator then anymore, but an advanced! :D

Nice though. ;)

I'll keep expanding my knowledge.

g4143 - March 22, 2014, 6:44 am

(Mar 22, 2014 06:19 AM)Niely Wrote: [\[CITE\]](#) ^Thanks for your help! :)

For goto and label:

But that's a really advanced/hard to understand code I think, it won't be an easy calculator then anymore, but an advanced! :D

Nice though. ;)

I'll keep expanding my knowledge.

The code you posted above has a bunch of typos.

Code #include <iostream>

Using namespace std;//uppercase U

Int main() { //uppercase I

}

Code Int x = 7;//uppercase I, C, I, C

Cout<<x<<endl;

Int y = 4;

```
Cout<<y<<endl;
```

Niely - March 22, 2014, 6:45 am

Yeah, I typed this tutorial in Word, I should delete the upper-cases.

g4143 - March 22, 2014, 6:52 am

(Mar 22, 2014 06:45 AM)Niely Wrote: [\[CITE\]](#) Yeah, I typed this tutorial in Word, I should delete the upper-cases.

You should never, never use a word processor for code, even if you are just going to post it on a forum it should be with ascii text editor or IDE.

Skyla Blue - March 22, 2014, 6:56 am

Bleh, I don't want to be mean, but all these "tutorials" written by complete beginners would actually do more harm than good, if someone decided to try and learn from them.

Masm4ever - March 22, 2014, 7:20 am

(Mar 22, 2014 05:06 AM)g4143 Wrote: [\[CITE\]](#) Could you explain why you need goto's in your program?

That's the latest fashion: Google [Apple GOTO](#) (over 10 Million hits)

Niely - March 22, 2014, 8:21 am

(Mar 22, 2014 06:56 AM)Skyla Blue Wrote: [\[CITE\]](#) Bleh, I don't want to be mean, but all these "tutorials" written by complete beginners would actually do more harm than good, if someone decided to try and learn from them.

Yeah, it indeed isn't the best way to learn. :)

But it could be interesting for people who are on the same 'level' of me.

So they know how this could be done approximately.

And a nice way to learn for me too. :P

(Mar 22, 2014 06:52 AM)g4143 Wrote: [\[CITE\]](#) You should never, never use a word processor for code, even if you are just going to post it on a forum it should be with ascii text editor or IDE.

I'll keep that in mind! Thank you! :)