

Lab 3

Variables and math operations

Objective:

To get acquainted with some C++ variable types and mathematical operations.

Background:

As you know, computers process data. Computer data is stored in *variables*. A variable is a place in the computer's memory that holds information.

Common types of variables in C++:

Name	Description
int	Holds integer numbers such as 1 or -37004
double	Holds decimal numbers such as 3.14159 and -0.001
char	Holds a single character such as 'A' or '\$' or '4'
string	Holds a series of characters such as "Emily" or "Programming is fun!"
bool	Holds true or false (more about this later)

Typecasting:

Sometimes you may want to *temporarily* change the type of a variable. A *typecast* changes the data type of one variable into another. The format is (dataType) value.

An example of typecasting: (double) myInt

This temporarily changes the integer variable myInt to the type of variable that can hold a decimal number.

You try it:

1. Beginning with a new "Hello World" project with an appropriate name, edit your code to look like the example below, and run the program:

```
#include <iostream>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```

    int i = 18, j, answer;
    j = 3;
    answer = i/j;
    cout << "int variables " << i << " / " << j << " = " << answer << endl; return 0;
}

```

Make sure your program prints **int variables 18 / 3 = 6** as you would expect.

2. Modify your program making the one change to the variable j shown below:

```

#include <iostream>
using namespace std;

int main()
{
    int i = 18, j, answer;
    j = 4;
    answer = i/j;
    cout << "int variables " << i << " / " << j << " = " << answer << endl; return 0;
}

```

What happened? Can you explain why?

Variable J was set to 4 so as a result what was printed out was "int variables 18 / 4 = 4". Since variable J was identified as an integer in the beginning of the function, the decimal points are not able to be calculated since integers only display solid numbers.

3. Now modify your program to match the following:

```

#include <iostream>
using namespace std;

int main()
{
    int i = 18, j, answer1;
    int answer2;
    j = 4;
    answer = i/j;
    answer2 = i%j;
    cout << "int variables " << i << " / " << j << " = " << answer1 << endl;
    cout << "int variables " << i << " % " << j << " = " << answer2 << endl;
}

```

```
    return 0;
}
```

% is called the *modulus* operator. Can you explain what it does?

The modulus operator tells you the remainder of the two numbers you are trying to divide.

4. Now edit your program so that your variables are doubles instead of ints, like this:

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

int main()
{
    double i = 18, j, answer1;
    double answer2;
    j = 4;
    answer = i/j;
    answer2 = i%j;
    cout << "double variables " << i << " / " << j << " = " << answer1 << endl; cout << "
    double variables " << i << " % " << j << " = " << answer2 << endl; return 0;
}
```

Notice that you get an error when you try to compile your program! The % operator (*modulus operator*) only makes sense for integer division. Why?

A division with remainder makes only sense on integers since numbers with decimals can always be done without a remainder.

To fix your program, **comment out** that line like this:

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

int main()
{
    double i = 18, j, answer1;
    double answer2;
    j = 4;
```

answer = i/j;

```

// answer2 = i%j;
cout << "double variables " << i << " / " << j << " = " << answer1 << endl; cout <<
"double variables " << i << " % " << j << " = " << answer2 << endl; return 0;
}

```

Now, what happened? Why? What value do you get for answer2?

I did not get an answer for answer2 because "answer2 = i%j;" was commented out meaning answer2 had no value anymore and therefore when it was told to print out "answer2" nothing would return.

5. Next we'll modify the program so that you can change the test numbers as you run your program. To do that you'll use the cin command. This program also demonstrates the ++ (increment) and -- (decrement) operators and typecasting.

```

#include <iostream>
using namespace std;

int main()
{
    int i = 18, j, answer1;
    int answer2;
    double answer3, answer4;

    cout << "Enter integer values for i and j: ";           // Get input from the user
    cin >> i >> j;
    answer1 = i/j;      //integer division
    answer2 = i%j;      //modulus operator

    cout << "\nint variables " << i << " / " << j << " = " << answer1 << endl; cout <<
    "int variables " << i << " % " << j << " = " << answer2 << endl<<endl;

    i++; //increment operator
    j--; //decrement operator
    cout << "i is now: " << i << ", j is now: " << j << endl << endl;

    answer3 = i/j;
    answer4 = (double)i/j;      //example of typecasting
    cout << "double answer3 = " << i << " / " << j << " = " << answer1 << endl;
    cout << "double answer4 = " << i << " / " << j << " = " << answer2 << endl;
}

```

```
    return 0;  
}
```

Run this program a few times with different inputs. Show your results here:

```
i = 5, j = 3  
int variable 5 / 3 = 1  
int variables 5 % 3 = 2
```

i is now: 6, j is now: 2

```
double answer3 = 6 / 2 = 3  
double answer4 = 6 / 2 = 3
```

```
i = 12, j = 16  
int variable 12 / 16 = 0  
int variables 12 % 16 = 12
```

i is now: 13, j is now: 15

```
double answer3 = 13 / 15 = 0  
double answer4 = 13 / 15 = 0.866667
```

```
i = 10, j = 5  
int variable 10 / 5 = 2  
int variables 10 % 5 = 0
```

i is now: 11, j is now: 4

```
double answer3 = 11 / 4 = 2  
double answer4 = 11 / 4 = 2.75
```

Explain what happens when two integers are divided:

When two integers are divided the result is the whole number of the dividend.

Explain what the modulus operator does:

The modulus gives me the remainder once the division has been completed.

Explain the function of the ++ and -- operators:

++ increases the value of a variable by 1, in this program i is being increased.

-- decreases the value of a variable by 1, in this program j is being decreased.

Explain why answer3 is different than answer4. If your answer3 is the same as answer4, run your program with different inputs until you get different values for answer3 and answer4:

Answer4 is different because variable i becomes a double meaning decimals were added which changed the results.

Problems:

Assume the following declarations have already been made:

```
int int1 = 17, int2 = 5, myInt;
```

```
double two = 2.0, three = 3.0, myDouble;
```

Find the value assigned to the give variable or indicate why the statement is not valid:

```
myInt = int1/int2; 3
```

```
myInt = int1 % int2; 2
```

```
myInt = int1/6 + 2; 4
```

```
myInt = 12 + (int2%6) - 4; 9
```

```
myDouble = three % two; error
```

```
int1++; 18
```

```
two--; 4
```

```
myInt = (int) three % 2; 1
```

```
myDouble = (double) int1/2.0; 8.5
```