

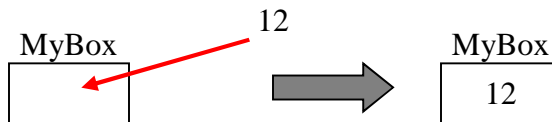
Unit 2: Variables – Integer and Float

1. Variables

We can consider *variables* as boxes. We can save money in our box; similarly, we can save numbers into variables. Every variable box has a name.

Variable name { MyBox }
[] a variable

Whenever we need, we can take and put the money in the box, similarly we can access the number in the box and put a new number.



In C++, we can declare `MyBox` variable by writing:

```
int MyBox;
```

Each variable has a type, that is, it can hold a certain type of information. The type of the information of the variable is given in the definition of the variable, right before its name. For example, in the definition above `int` specifies its type, that is, variable `MyBox` is an integer type, and can only contain integer values.

We can declare a variable that can hold decimal values as follows:

```
float MyFloatBox;
```

As an example, `MyFloatBox` can contain a decimal value such as 12.123 as well as an integer value.

Declaring a variable is like buying a new box from the store.

Note: We have to be careful about variable names; they do not contain space and they are case sensitive. `MyBox` and `myBox` are considered as two different variables.

Note: The first character of a variable name has to be a letter, the rest can be letters, numbers, or underscore.

Example. 'My_3rd_Box' is a valid variable name whereas '3rdMyBox' not since it starts with a number.

In C++, after we define MyBox, we can put money in MyBox.

Example. We will buy a box that has the name 'nicebox' and put \$10 in that box.

Open a new file, write the code below, name it as 'nicebox.cpp', save it on the desktop and run it.

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

int main()
{
    int nicebox;
    nicebox = 10;
}
```

Note: Equal sign (=) in C++, is not the same as the equal sign in math. In math, equal sign means '*the results on both sides of the equality are equal*'. On the other hand, in C++ equal sign means '*put the number on the right side into the box on the left side*'.

Example. In C++, 'Mybox = 4' doesn't mean 'Mybox and 4 are equal'; it means 'put 4 into Mybox'.

If we try to put a decimal number into an integer type variable then the decimal part of the number is ignored.

Example. 'Mybox = 4.78' puts 4 in Mybox. Same thing happens in the following code.

```
float number;
int Mybox;
number = 4.78;
Mybox = number;
```

The value in Mybox becomes 4.

We can declare more than one variable of the same type in one line. For example,

```
int num1, num2, num3;
```

is the same as;

```
int num1;
int num2;
int num3;
```

While defining variables, we can also set their initial values. For instance, the following line initializes `num1` and `num3` with values 3 and 123.77 respectively.

```
float num1 = 3, num2, num3 = 123.77;
```

2. Writing Numbers in the Variables to the Screen

Example. Consider the following example.

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

int main()
{
    int nicebox;
    nicebox = 10;
}
```

Question: When you run that program, you did not see anything on the screen. Why?

Suppose we would like to write the money in ‘nicebox’ to the screen.

Open a new file, write the code below, name it as ‘nicebox2.cpp’, save it on the desktop and run it.

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

int main()
{
    int nicebox;
    nicebox = 10;
    cout << nicebox << endl;
}
```

Question: What did you see on the screen when you run the program?

Exercise 1: What will you see on the screen after running the program below?

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

int main()
{
    float nicebox;
    nicebox = 10.3;
    nicebox = 7.51;
    cout << nicebox << endl;
}
```

Exercise 2: What will you see on the screen after running the program below?

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

int main()
{
    int nicebox;
    nicebox = 123;
    float anotherbox;
    anotherbox = 21.186;
    cout << anotherbox << endl;
    cout << nicebox << endl;
}
```

We can output multiple expressions with `cout`. For instance,

```
int secondsInADay = 86400;
cout << "There are " << secondsInADay << " seconds in one day.";
```

will output:

```
There are 86400 seconds in one day.
```

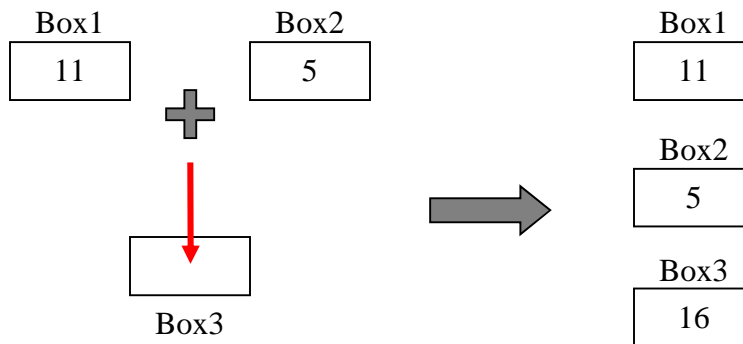
In C++, it doesn't matter if an expression is in one line, or separated into multiple lines. For example, the following codes are the same:

```
cout << "There are " << secondsInADay << " seconds in one day.";
```

```
cout << "There are "
    << secondsInADay <<
    " seconds in one day.";
```

3. Arithmetic Operations on Variables

Suppose we have to three boxes two of which have some money in them. Then we can get money in these two boxes and put all the money into the third box. Similarly, suppose we have three variables named `Box1`, `Box2`, and `Box3` where `Box1` and `Box2` contain the numbers 11 and 5 respectively. We can add the number in `Box1` and `Box2` and put the sum into `Box3`. However, we do not lose the numbers in the boxes; they also remain the same.



Example. Suppose you have two three boxes, Box1, Box2, and Box3. Box1 and Box2 contain 11 and 5 respectively. Write a C++ program that adds the numbers in two boxes, puts it in Box3, and outputs Box3 to the screen.

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

int main()
{
    int Box1;
    int Box2;
    int Box3;
    Box1 = 11;
    Box2 = 5;
    Box3 = Box1 + Box2;
    cout << Box3 << endl;
}
```

Output.

16

We can use the arithmetic operators, *addition* (+), *subtraction* (–), *division* (/), *multiplication* (*) and *parenthesis* ‘()’ in the same way we use in math.

Example. Suppose we have a box called arithbox. The following C++ program puts the result of ‘213 * 4 – 10’ into arithbox and writes the value of arithbox to the screen.

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

int main()
{
    int arithbox;
    arithbox = 213 * 4 – 10;
    cout << arithbox << endl;
}
```

Output.

842

Example. Suppose we have two boxes, box1 and box2. The following C++ program puts the result of ‘(2007 + 1881) / 2’ into box1 then it writes box1 to the screen. Next, it multiplies box1 by 3, adds 25 and puts the result into box2. Finally, it writes box2 to the screen.

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

int main()
{
    int box1;
    int box2;
    box1 = (207 + 181) / 2;
    cout << box1 << endl;
    box2 = box1 * 3 + 25;
    cout << box2 << endl;
}
```

Output.

```
194
607
```

Exercise 3: Write a program that puts the result of ‘363 / 3’ into box1. Next, it adds 2 to box1, multiply by 11 and puts the result into box2. Finally, it writes box2 then box1 to the screen.

Division and Remainder Operations

In C++, division operator is ‘/’ whereas remainder is ‘%’ for integers.

Example. ‘7 / 3’ is equal to 2. ‘7 % 3’ is equal to 1 since the remainder of division of 7 by 3 is 1.

For float type, the division will result in regular decimal value. For instance, ‘5 / 2’ is equal to 2.5.

When an arithmetic operation consists of both integer and decimal variables or numbers, the result is decimal. The following code;

```
cout << (4 + 5.66) / 3 << endl;
```

outputs 3.22 on the screen.

4. Reading Numbers from Keyboard

In C++, we can read numbers from keyboard and put the number we read into the variable boxes using ‘cin’ command.

Example. The following C++ program puts two numbers read from keyboard into `box1` and `box2`. Then it puts the half of the sum of `box1` and `box2` into `box3` and it writes `box3` to the screen.

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

int main()
{
    int box1;
    int box2;
    int box3;
    cin >> box1;
    cin >> box2;
    box3 = (box1 + box2) / 2;
    cout << box3 << endl;
}
```

Exercise 4: Write a program that reads two numbers from the keyboard and put them into `box1` and `box2`. Then it puts the multiplication of `box1` and `box2` into `box3` and it writes `box3` to the screen.

Similarly, we can read multiple variables with `cin`. For instance,

```
int number1, number2, number3;
cin >> number1 >> number2 >> number3;
```

reads three values into `number1`, `number2`, and `number3` from keyboard in the same order.