Al-Mustaqbal University

College of Engineering & Technology

Biomedical Engineering Department

Computer

Lab 2
Variables and Operators

PhD. BEng. Ahmed Hasan Janabi
PhD in Cybersecurity
Email: Ahmed.Janabi@uomus.edu.iq

Learn how to use, Variables and Operators (Assignment, Arithmetic operators, Relational and Logical operators, Bitwise Operators, Increment and decrement, Cast operator, and Conditional operator), Precedence of operators.

C++ Variables

In C++, there are different types of variables (defined with different keywords), for example:

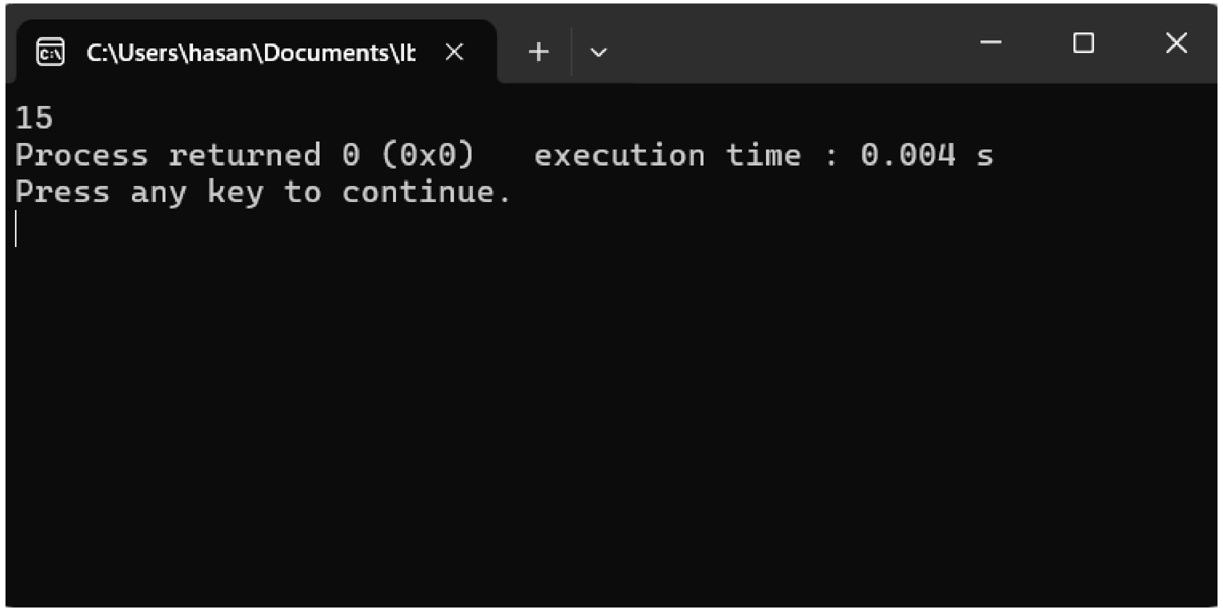
- int - stores integers (whole numbers), without decimals, such as 123 or -123
- double - stores floating point numbers, with decimals, such as 19.99 or -19.99
- char - stores single characters, such as 'a' or 'B'. Char values are surrounded by single quotes
- string - stores text, such as "Hello World". String values are surrounded by double quotes
- bool - stores values with two states: true or false

To create a variable, specify the type and assign it a value:

```
type variableName = value;
```

❑ Write the following c++ program to print the value of a variable?

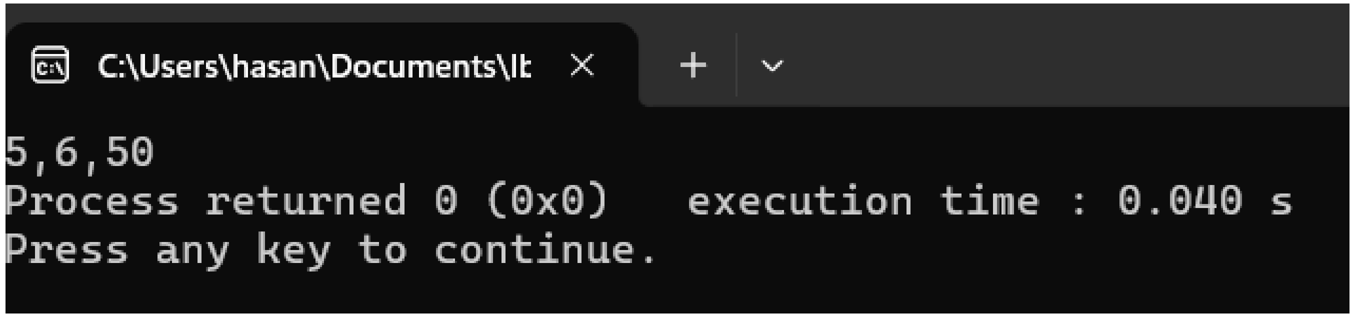
```cpp
#include <iostream>
using namespace std;

int main() {
  int myNum = 15;
  cout << myNum;
  return 0;
}
```



```
15
Process returned 0 (0x0)   execution time : 0.004 s
Press any key to continue.
```

❑ To declare more than one variable of the same type, use a comma-separated list:

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

int main() {
  int x = 5, y = 6, z = 50;
  cout << x << "," << y << "," << z;
  return 0;
}
```

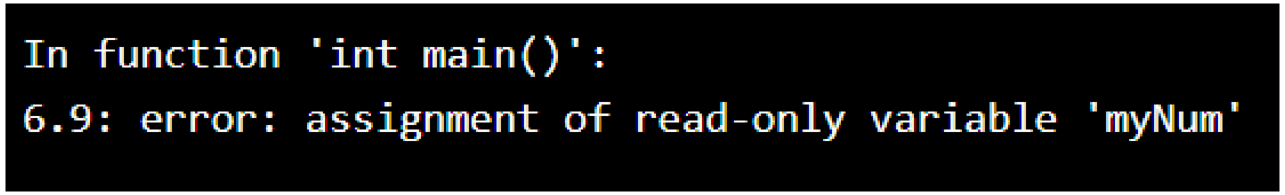


```
5,6,50
Process returned 0 (0x0)   execution time : 0.040 s
Press any key to continue.
```

C++ Constants

- When you do not want others (or yourself) to change existing variable values, use the const keyword (this will declare the variable as "constant", which means unchangeable and read-only):

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

int main() {
  const int myNum = 15;
  myNum = 10;
  cout << myNum;
  return 0;
}
```

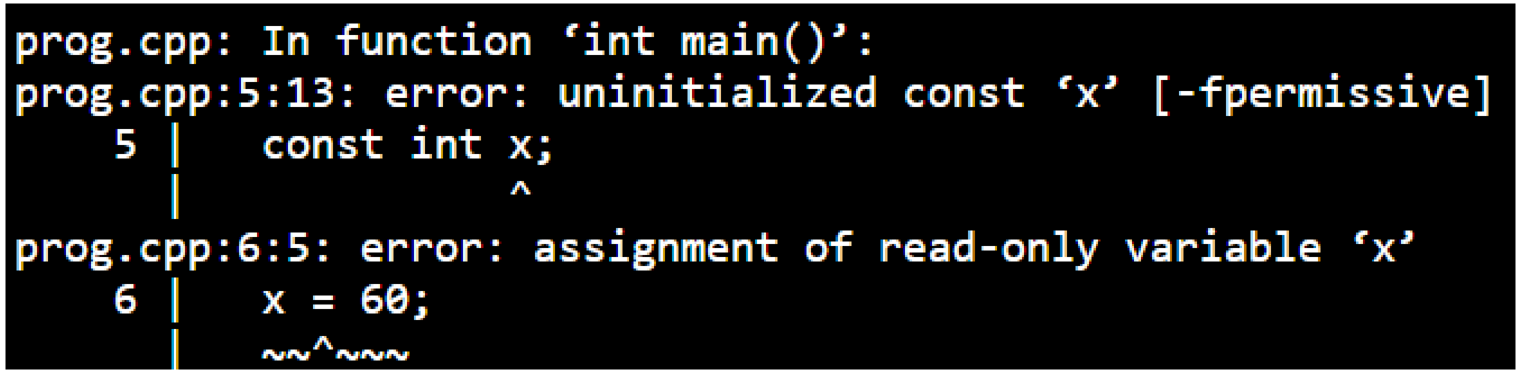


```
In function 'int main()':
6.9: error: assignment of read-only variable 'myNum'
```

C++ Constants

- ❑ Note: When you declare a constant variable, it must be assigned with a value:

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

int main() {
  const int x;
  x = 60;
  cout << x;
  return 0;
}
```



```
prog.cpp: In function ‘int main()’:
prog.cpp:5:13: error: uninitialized const ‘x’ [-fpermissive]
    5 |   const int x;
      |             ^
prog.cpp:6:5: error: assignment of read-only variable ‘x’
    6 |   x = 60;
      |   ~~^~~~
```

C++ User Input

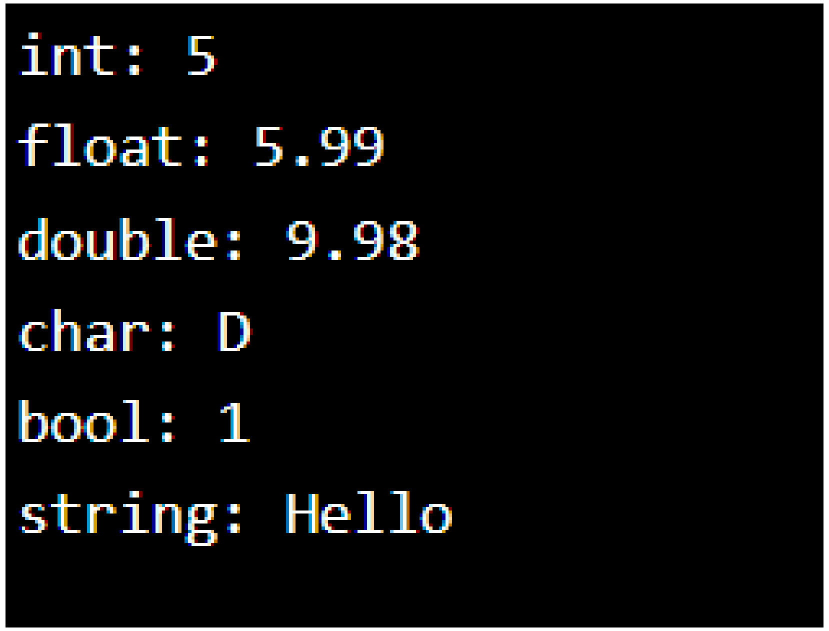
- You have already learned that cout is used to output (print) values. Now we will use cin to get user input.
- cin is a predefined variable that reads data from the keyboard with the extraction operator (>>).
- In the following example, the user can input a number, which is stored in the variable x. Then we print the value of x:

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

int main() {
  int x;
  cout << "Type a number: "; // Type a number and press enter
  cin >> x; // Get user input from the keyboard
  cout << "Your number is: " << x;
  return 0;
}
```

Data Types in C++

- ❑ Data types define the type of data a variable can hold, for example an integer variable can hold integer data, a character type variable can hold character data etc.

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

int main () {
  // Creating variables
  int myNum = 5;               // Integer (whole number)
  float myFloatNum = 5.99;     // Floating point number
  double myDoubleNum = 9.98;   // Floating point number
  char myLetter = 'D';         // Character
  bool myBoolean = true;       // Boolean
  string myString = "Hello";   // String

  // Print variable values
  cout << "int: " << myNum << "\n";
  cout << "float: " << myFloatNum << "\n";
  cout << "double: " << myDoubleNum << "\n";
  cout << "char: " << myLetter << "\n";
  cout << "bool: " << myBoolean << "\n";
  cout << "string: " << myString << "\n";

  return 0;
}
```



```
int: 5
float: 5.99
double: 9.98
char: D
bool: 1
string: Hello
```

C++ Operators

- ❑C++ divides the operators into the following groups:
  1. Arithmetic operators
  2. Assignment operators
  3. Comparison operators
  4. Logical operators
  5. Bitwise operators

1- Arithmetic operators

```cpp
#include <iostream>
using namespace std;
int main()
{
    int num1 = 240;
    int num2 = 40;
    cout<<"num1 + num2: "<<(num1 + num2)<<endl;
    cout<<"num1 - num2: "<<(num1 - num2)<<endl;
    cout<<"num1 * num2: "<<(num1 * num2)<<endl;
    cout<<"num1 / num2: "<<(num1 / num2)<<endl;
    cout<<"num1 % num2: "<<(num1 % num2)<<endl;
    return 0;
}
```

Output:

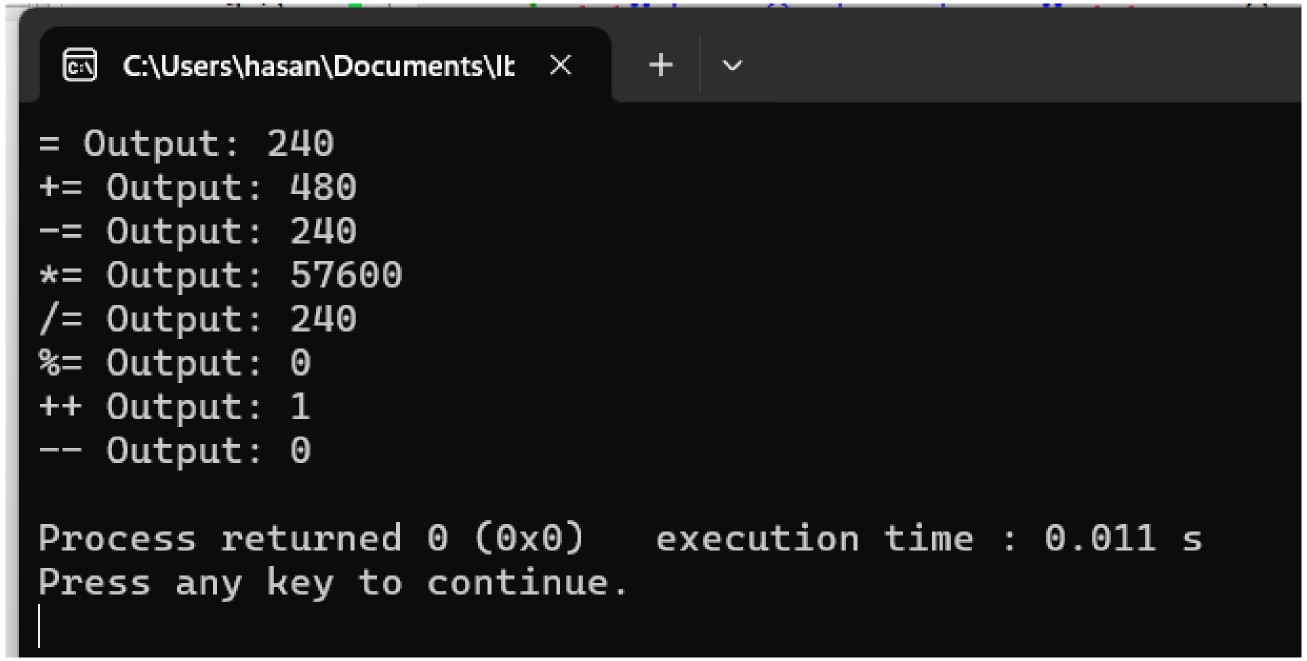
```
num1 + num2: 280
num1 - num2: 200
num1 * num2: 9600
num1 / num2: 6
num1 % num2: 0
```

2- Assignment Operators

- ❑ Assignments operators in C++ are: =, +=, -=, *=, /=, %=
- ➢ **num2 = num1** would assign value of variable num1 to the variable.
- ➢ **num2+=num1** is equal to num2 = num2+num1
- ➢ **num2-=num1** is equal to num2 = num2-num1
- ➢ **num2*=num1** is equal to num2 = num2*num1
- ➢ **num2/=num1** is equal to num2 = num2/num1
- ➢ **num2%=num1** is equal to num2 = num2%num1
- ➢ **++nm1** is equal to nm1 = nm1+1
- ➢ **--nm1** is equal to nm1 = nm1-1

Example of Assignment Operators

```cpp
#include <iostream>
using namespace std;
int main(){
    int num1 = 240;
    int num2 = 40;
    num2 = num1;
    cout<<"= Output: "<<num2<<endl;
    num2 += num1;
    cout<<"+= Output: "<<num2<<endl;
    num2 -= num1;
    cout<<"-= Output: "<<num2<<endl;
    num2 *= num1;
    cout<<"*= Output: "<<num2<<endl;
    num2 /= num1;
    cout<<"/= Output: "<<num2<<endl;
    num2 %= num1;
    cout<<"%= Output: "<<num2<<endl;
    ++num2;
    cout<<"++ Output: "<<num2<<endl;
    --num2;
    cout<<"-- Output: "<<num2<<endl;
    return 0;
}
```



```
= Output: 240
+= Output: 480
-= Output: 240
*= Output: 57600
/= Output: 240
%= Output: 0
++ Output: 1
-- Output: 0

Process returned 0 (0x0)   execution time : 0.011 s
Press any key to continue.
```

3- Comparison operators

- ❑ We have six relational operators in C++: ==, !=, >, <, >=, <=
- ➢ == returns true if both the left side and right side are equal
- ➢ != returns true if left side is not equal to the right side of operator.
- ➢ > returns true if left side is greater than right.
- ➢ < returns true if left side is less than right side.
- ➢ >= returns true if left side is greater than or equal to right side.
- ➢ <= returns true if left side is less than or equal to right side.

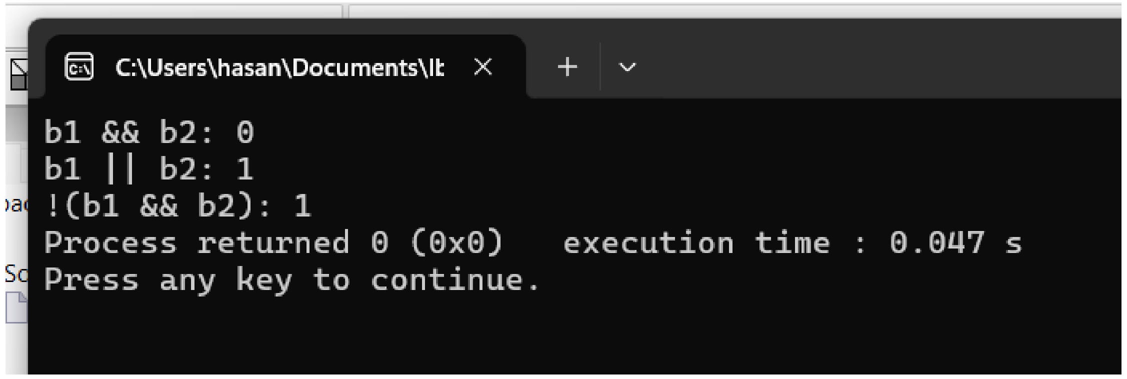
```
#include <iostream>
using namespace std;

int main() {
  int x = 5;
  int y = 3;
  cout << (x > y); // returns 1 (true) because
5 is greater than 3
  return 0;
}
```

4- Logical Operators

- ❑ Logical Operators are used with binary variables. They are mainly used in conditional statements and loops for evaluating a condition.
- ➢ Logical operators in C++ are: &&, ||, !
- ➢ Let’s say we have two boolean variables b1 and b2.
- ➢ **b1&&b2** will return true if both b1 and b2 are true else it would return false.
- ➢ **b1||b2** will return false if both b1 and b2 are false else it would return true.
- ➢ **!b1** would return the opposite of b1, that means it would be true if b1 is false and it would return false if b1 is true.

Example of Logical Operators

```cpp
#include <iostream>
using namespace std;
int main(){
    bool b1 = true;
    bool b2 = false;
    cout<<"b1 && b2:  "<<(b1&&b2)<<endl;
    cout<<"b1 || b2:  "<<(b1||b2)<<endl;
    cout<<"!(b1 && b2):  "<<!(b1&&b2);
    return 0;
}
```



```
b1 && b2: 0
b1 || b2: 1
!(b1 && b2): 1
Process returned 0 (0x0)   execution time : 0.047 s
Press any key to continue.
```

Homework

Create a Currency Converter

1. Declare variables to store the amount in US dollars and the conversion rate to IQD.
2. Prompt the user to enter the amount in US dollars.
3. Perform the conversion using the formula Euros=Dollars×Conversion IQD=Dollars×Conversion Rate.
4. Output the converted amount in IQD.
5. Assume a conversion rate, or you can make it more dynamic by also asking the user for the current conversion rate.

DO YOUR BEST
AS
YOU ARE THE BEST