

مدخل إلى الخوارزميات والبرمجة هندسة الميكاترونكس سنة أولى

2023-2024

مدرس المقرر: د. عيسى الغنام

Lecture No. 2

C++ Data Types And Variables

1-2 Data types:

Variables are containers for storing data values. In C++, there are different types of variables (defined with different keywords), for example:

	TYPE NAME	MEMORY USED	NOTE
numerical	short (also called short int)	2 bytes	-32,768 to 32,767
	int	4 bytes	-2,147,483,648 to 2,147,483,647 int is a type for storing integer (whole) numbers. -2^{31} to $2^{31}-1$
	long (also called long int)	4 bytes	
	float	4 bytes	The range of float type in the C++ can represents values ranging from approximately: -3.4×10^{38} and 3.4×10^{38}
	double	8 bytes	The double type stores floating point (decimal) numbers.
	long double	12 bytes	
	unsigned	4 bytes	0 to $4294967295=2^{32}-1$
text	char	1 byte	All ASCII characters (Can also be used as an integer type)

	string	24 bytes	character sequences wrapped in double quotes (e.g., "Hello World!").
logical	bool	1 byte	true , false The bool type stores Boolean values of true or false.

- يجب التصريح عن كل المعطيات قبل استخدامها في البرنامج.
- إعطاء المعطيات قيم ابتدائية في تعليمة التصريح

In short:

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

2-2 Variables Names:

The general rules for naming variables are:

- Names can contain letters, digits and underscores
- Names must begin with a letter or an underscore (_). It cannot begin with a number.
- Names are case sensitive (myVar and myvar are different variables)
- Names cannot contain whitespaces.
- Reserved words (like C++ keywords, such as int) cannot be used as names
- Names Cannot have a symbol, or special characters like !, #, %, etc. expect (_ \$)

3-2 Declaring a single variable:

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

Syntax: `data_type variableName = value;`

Where **data_type** is one of C++ types (such as int), and **variableName** is the name of the variable (such as **x** or **myName**). The equal sign is used to assign values to the variable. To create a variable that should store a number, look at the following example:

<pre>int myNum = 15; cout << myNum; //Here, myNum is a variable of the int data //type, and we have assigned an integer //value 15 to it.</pre>	<pre>int myNum; myNum = 15; cout << myNum;</pre>	<pre>int myNum = 15; // myNum is 15 myNum = 10; // Now myNum is 10 cout << myNum; // Outputs 10</pre>
--	--	---

```
int myNum = 5;          // Integer (whole number without decimals)
double myFloatNum = 5.99; // Floating point number (with decimals)
char myLetter = 'D';    // Character
string myText = "Hello"; // String (text)
bool myBoolean = true;  // Boolean (true or false)
```

4-2 Declaring multiple variables:

Syntax: *data_type variableName1, variableName2, variableName3;*

<pre>int x = 5, y = 6, z = 50; cout << x + y + z;</pre>	<pre>int x, y, z; x = y = z = 50; cout << x + y + z;</pre>
---	--

المثال الأول : أنماط المتحولات وحجمها:

```
#include <iostream>

using namespace std;

int main()
{
    int x=2147483647;

    //(-2^31)=-2147483648 to (2^31)-1=+2147483647

    cout << "int:" << sizeof(x) << "byte\r" << x << endl << "-----\n";
```

```

////////////////////////////////////
long z=-2147483648;

//-[((2^31))]=-2147483648 to [(2^31)-1]=+2147483647

cout <<"long:"<< sizeof(z)<<"byte\t"<<z<<endl<<"-----\n";

////////////////////////////////////

short y=-32768;

//-[((2^16)/2)]=-32768 to [(2^16)/2)-1]=+32767

cout <<"short:"<< sizeof(y)<<"byte\t"<<y<<endl<<"-----\n";

////////////////////////////////////

float r=2.147;

cout <<"float:"<< sizeof(r)<<"byte\t"<<r<<endl<<"-----\n";

////////////////////////////////////

double h=500.14754;

cout <<"double:"<< sizeof(h)<<"byte\t"<<h<<endl<<"-----\n";

////////////////////////////////////

long double j=500.14754;

cout <<"long double:"<< sizeof(j)<<"byte\t"<<j<<endl<<"-----\n";

////////////////////////////////////

char q='srrrh';

cout <<"char:"<< sizeof(q)<<"byte\t"<<q<<endl<<"-----\n";

////////////////////////////////////

bool p=0;//p=1//p=0

cout <<"bool:"<< sizeof(p)<<"byte\t"<<p<<endl<<"-----\n";

////////////////////////////////////

string ss="sggggnnnnnnnnnnnnnnnnnnnnnnnnnnnnnrrr4554555hgjjjfh";

cout <<"string:"<< sizeof(ss)<<"byte\t"<<ss<<endl<<"-----\n";

return 0;

}

```

 "E:\0.SCIENCE\0.courses\1.programming languages\c++\0 course\bycodeblocks\course 2023\1

```
int:4byte      2147483647
-----
long:4byte     -2147483648
-----
short:2byte    -32768
-----
float:4byte     2.147
-----
double:8byte    500.148
-----
long double:12byte  500.148
-----
char:1byte      h
-----
bool:1byte      0
-----
string:24byte   sggggnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnrrr4554555hgjjjfh
```

```
Process returned 0 (0x0)   execution time : 0.750 s
Press any key to continue.
```

المثال الثاني : أسماء المتحولات

```
#include <iostream>

using namespace std;

int main()

{

/* أسماء المتحولات */

// int 1x=5;

int x2=3;

// int float=6

float _x=88;

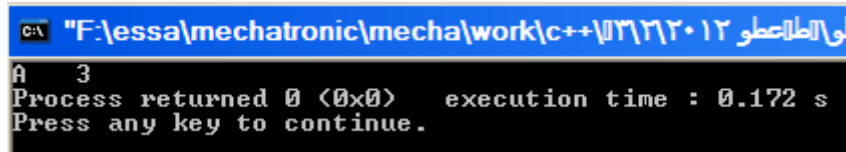
// double h#y=6;

double h_y=6;

char X2='A';

cout<<X2<<" "<<x2;
```

```
return 0;
}
```



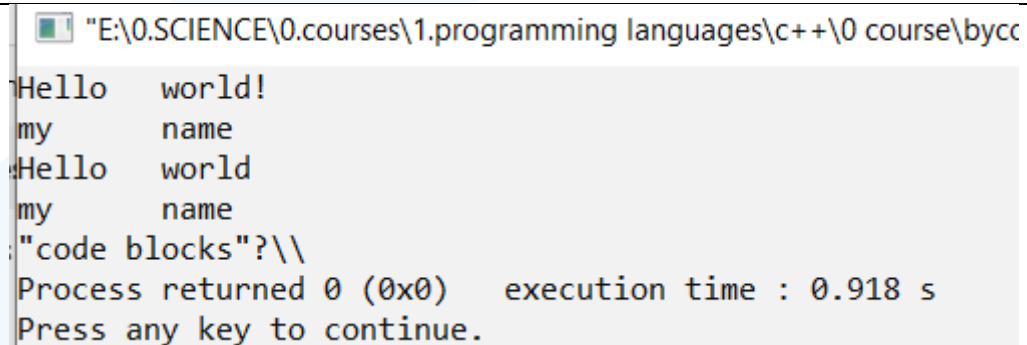
Process returned 0 (0x0) execution time : 0.172 s
Press any key to continue.

المثال الثالث : تتابعات الهروب escape sequences

```
#include <iostream>

using namespace std;

int main()
{
    cout << "Hello\tworld!\nmy\tname\a" << endl;
    cout << "Hello\tworld!\b\nmy\tname\a" << endl;
    cout << "\"code blocks\""?\\\\";
    return 0;
}
```



Process returned 0 (0x0) execution time : 0.918 s
Press any key to continue.

1. \ – Backslash: Represents a single backslash (\) in the output.
2. \' – Single Quote: Represents a single quote (') character.
3. \" – Double Quote: Represents a double quote (") character.

4. \n – Newline: Inserts a line break.
5. \t – Horizontal Tab: Inserts a horizontal tab space.
6. \r – Carriage Return: Moves the cursor to the beginning of the current line.
7. \b – Backspace: Moves the cursor one position back.
8. \f – Form Feed: Advances the cursor to the next page or form feed.
9. \v – Vertical Tab: Inserts a vertical tab space.
10. \a – Alert: Produces a system alert or beep sound.
11. \? – Question Mark: Represents a question mark (?) character.
12. \ooo – Octal Number: Represents a character using its octal value, where 'ooo' is a 1- to 3-digit octal number.
13. \xhh – Hexadecimal Number: Represents a character using its hexadecimal value, where 'hh' is a 1- to 2-digit hexadecimal number.

انتهت المحاضرة