Advanced Programming (C++)

BY

Dr. EMAD SAMI

http://www.bu.edu.eg/staff/emadattwa3

Course Chapters

1. Introduction
2. Variables and Constants
3. Expressions and Statements
4. Loops and Decisions
5. Functions
6. Arrays and Strings
7. Pointers
8. Miscellaneous

2. Variables and Constants

Chapter Objectives:

2-1 What is the Variable in C++ Language?

2-2 Declare (Define) a Variable inside C++ Program

2-3 Rules (Constrains) for Naming a Variable

2-4 Integer Variable (**int**)

2-5 Character Variable (**char**)

2-6 Float Variable (**float**)

2-7 Naming constants and Notes

2-8 Variable Type (**long int**)

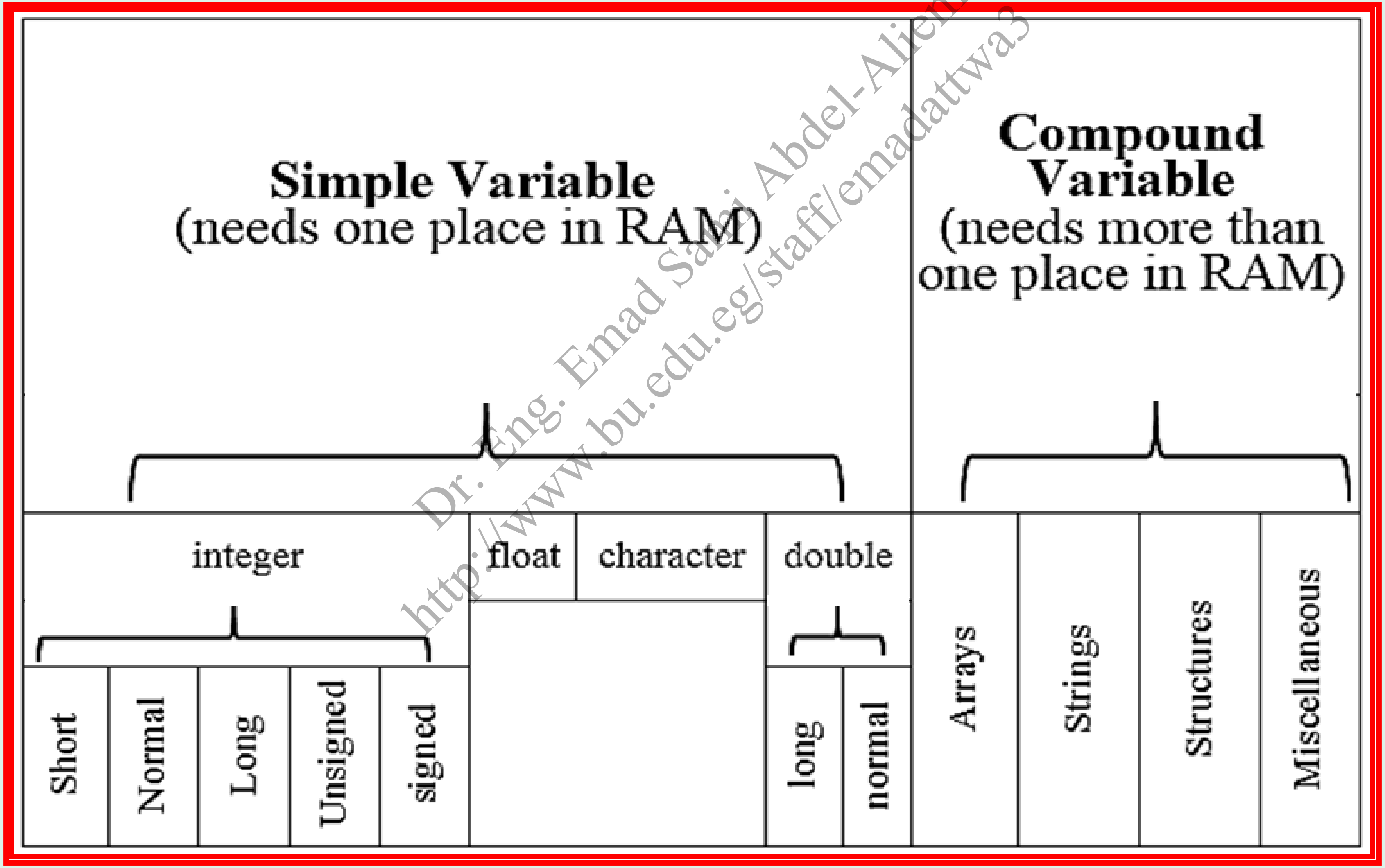
2-9 Table of Variable Type and Its Size

2-10 Automatic Data Conversion

2-11 Assignment (2)

2-1 What is the Variable in C++ Language?

- The variable is a place in the computer memory to store a value and from which you can later retrieve that value.
- There are two main types of variables in C++ program:



Simple Variable
(needs one place in RAM)

integer

Short

Normal

Long

Unsigned

signed

float

character

double

long

normal

Compound Variable
(needs more than one place in RAM)

Arrays

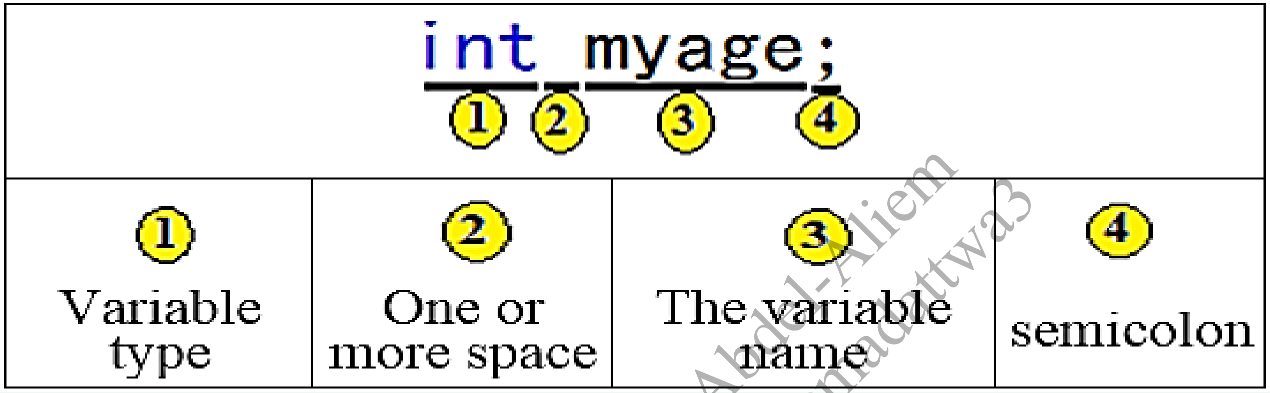
Strings

Structures

Miscellaneous

2-2 Declare or define a Variable inside C++ Program

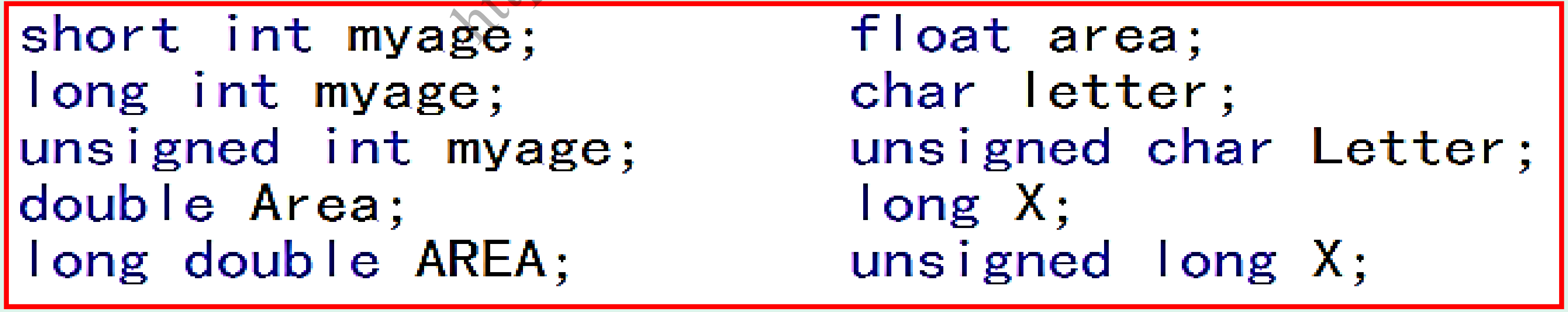
- Inside the **main** function, declare a variable is as follows:



```
int myage;
```

1	2	3	4
Variable type	One or more space	The variable name	semicolon

- Declare a variable is to determine a place in the RAM for that variable and it should have a name shows the purpose of its usage.
- Also, we can declare (or define) the variable in any place in the code, but this before using the variable or during using it.
- Examples for declare a variable:



```
short int myage;              float area;
long int myage;               char letter;
unsigned int myage;           unsigned char Letter;
double Area;                  long X;
long double AREA;             unsigned long X;
```

2-3 Rules (or Constrains) for Naming a Variable

- You **can** use the following at naming a variable:

1. Characters A through Z and also a through z.
2. Digit characters 0 through 9, which can be used in any position except the first of a variable name.
3. The underscore character (_)

*Examples for **valid** variable names:*

stop_sign Loop3 myAge

2-3 Rules (or Constrains) for Naming a Variable …

- You **can’t** use the following at naming a variable:

1. A variable name can’t contain any C++ arithmetic signs.
2. A variable name can’t contain any dots.
3. A variable name must not begin with numbers.
4. A variable name can’t contain any apostrophes.
5. A variable name can’t contain any other special symbols such as ?, @, #, &, $, and so on.
6. A variable name can’t contain any C++ keywords, or a name from the standard C library functions.

*Example of **invalid** variable names;*

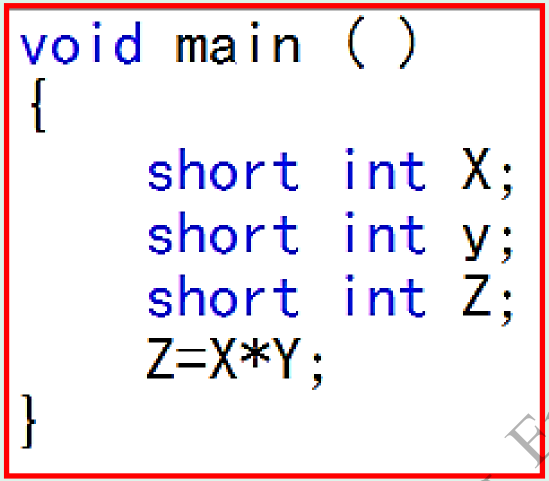
While 4flags sum.result

method*4 what_size? char

2-3 Rules (or Constrains) for Naming a Variable …

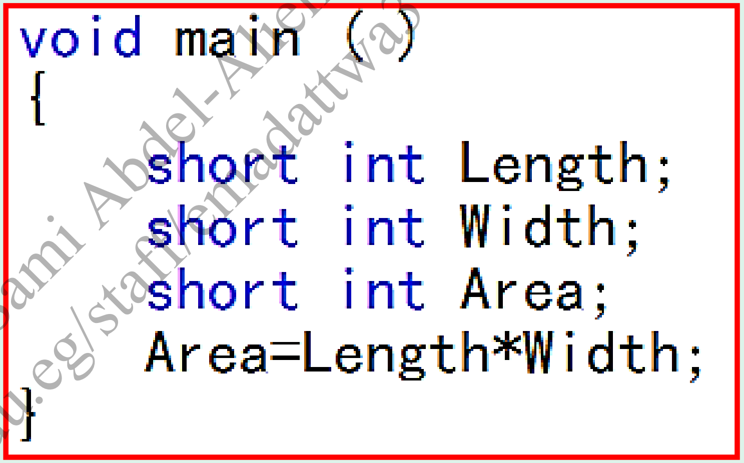
- For naming a variable use a certain meaningful words. Note the following code: *Clearly, Example no. 2 is easier to understand.*

Example no. 1



```
void main ( )
{
      short int X;
      short int y;
      short int Z;
      Z=X*Y;
}
```

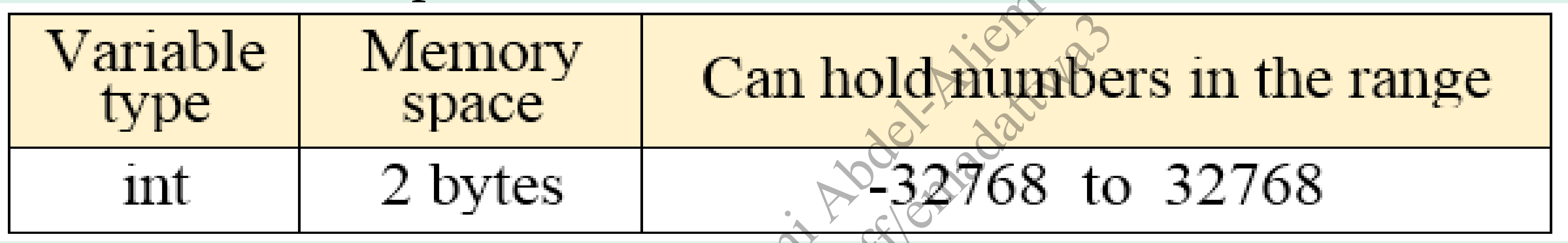
Example no. (2)



```
void main ( )
{
      short int Length;
      short int Width;
      short int Area;
      Area=Length*Width;
}
```

2-4 Integer Variable (int)

- Are represent the integer numbers like 1, 5, -35, …….100.
- Integer numbers may be positive or negative. But, they don't have a fractional part.



Variable type	Memory space	Can hold numbers in the range
int	2 bytes	-32768 to 32768

- For example, the number 50000 is illegal integer number to be used with **int** variable.
- What does mean of ***assign value*** to a variable?

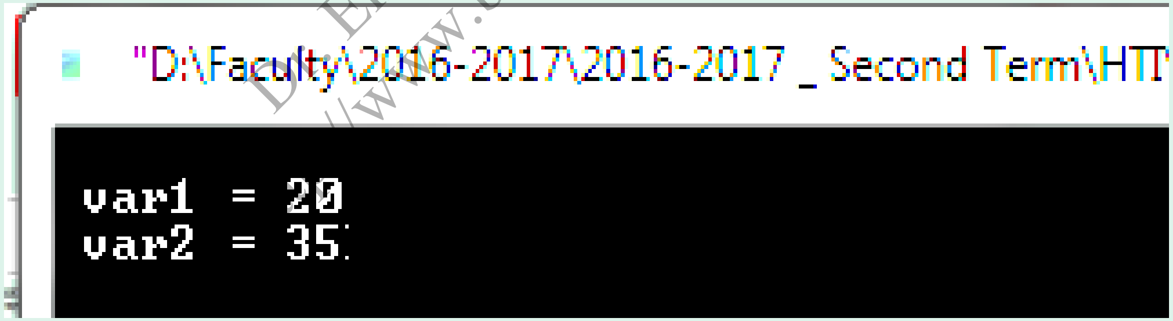
answer: give values (numbers or characters) to the variable.

int X=20; means the value **20** is assigned (or given) to the **int** variable **X** by using the sign (**=**).

2-4 Integer Variable (int) …

- Here is a program that define (declare) and uses several variables of type **int**.

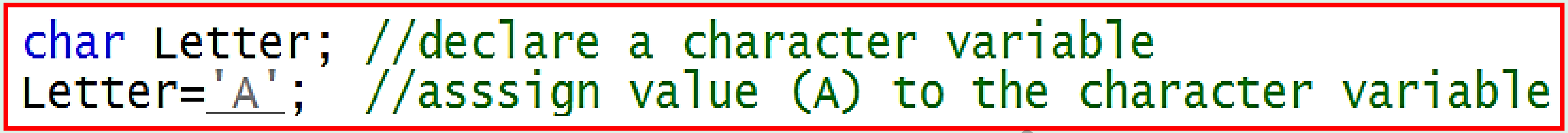
```
//int.cpp
//Demonstrate Integer Variables
#include <iostream.h>
void main ( )
{
   short int var1; //define a variable var1
   short int var2; //define a variable var2
   var1=20;        //assign value 20 to variable var1
   var2=var1+15;   //assign value (var1+15) to variable var2
   cout<<"\n var1 = "<<var1; //print value of var1 to the screen
   cout<<"\n var2 = "<<var2; //print value of var2 to the screen
}
```



```
 var1 = 20
 var2 = 35
```

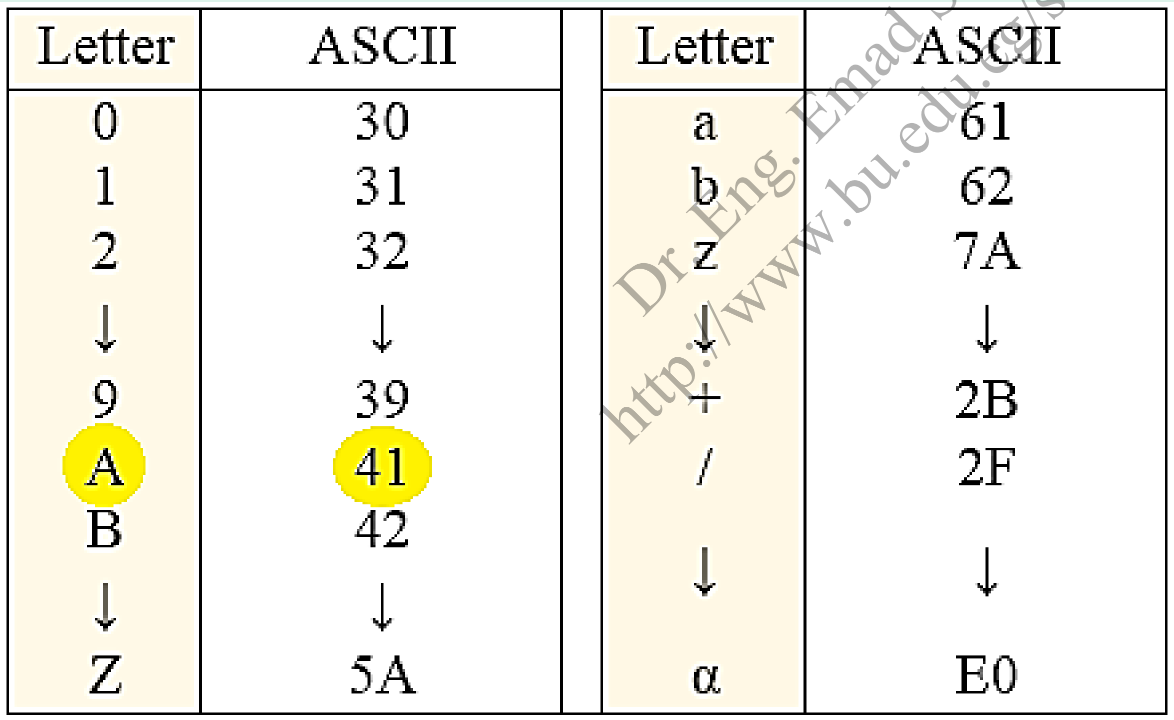
2-5 Character Variable (char)

- To declare and assign the character variable, see the picture:



```
char Letter; //declare a character variable
Letter='A';  //asssign value (A) to the character variable
```

- The character variable (type **char**) has a memory size of 1 byte.
- The character variable can be interpreted or equivalent to a small number (0 up to 255) or as a letter denoted by its ASCII code.
- ASCII = American Standard Code for Information Interchange.

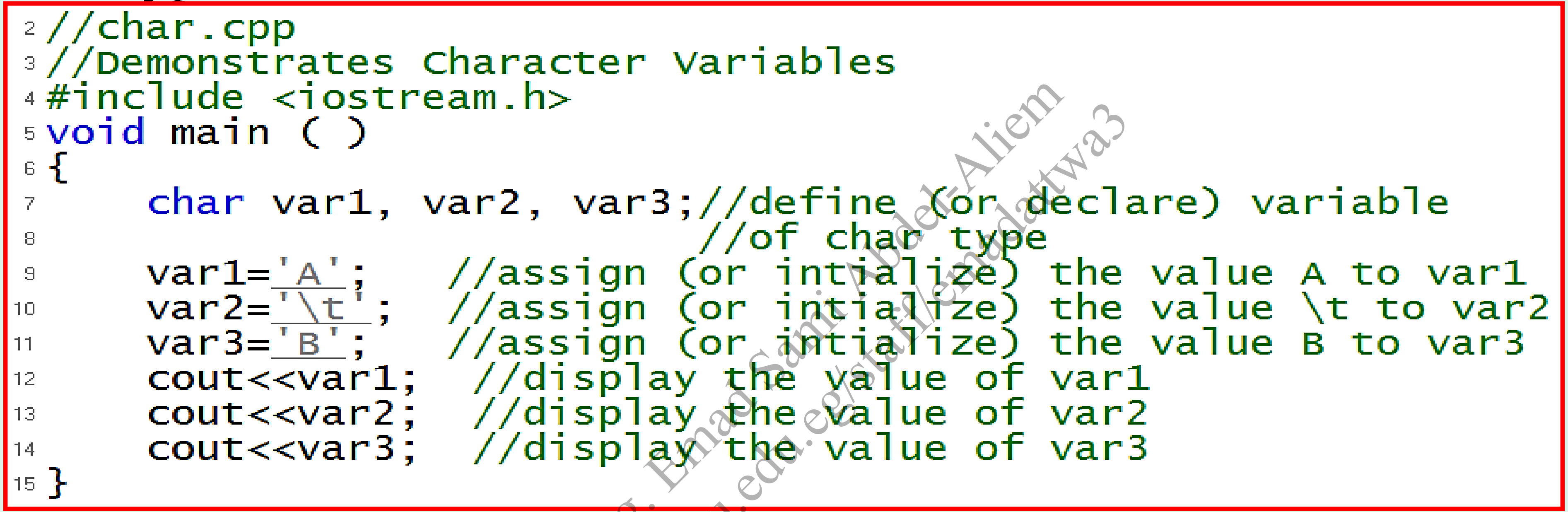


Letter	ASCII		Letter	ASCII
0	30		a	61
1	31		b	62
2	32		z	7A
↓	↓		↓	↓
9	39		+	2B
A	41		/	2F
B	42			
↓	↓		↓	↓
Z	5A		α	E0

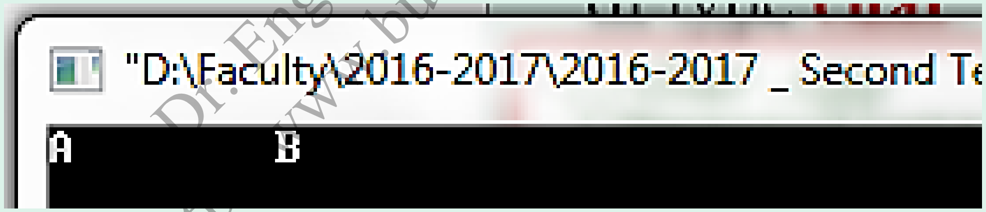
For example:
char A has a size in the memory equal to the size of the number **41**.

2-5 Character Variable (**char**) …

- Here is a program that define (declare) and uses several variables of type **char**.



```
//char.cpp
//Demonstrates Character Variables
#include <iostream.h>
void main ( )
{
    char var1, var2, var3;//define (or declare) variable
                          //of char type
    var1='A';   //assign (or intialize) the value A to var1
    var2='\t';  //assign (or intialize) the value \t to var2
    var3='B';   //assign (or intialize) the value B to var3
    cout<<var1;  //display the value of var1
    cout<<var2;  //display the value of var2
    cout<<var3;  //display the value of var3
}
```

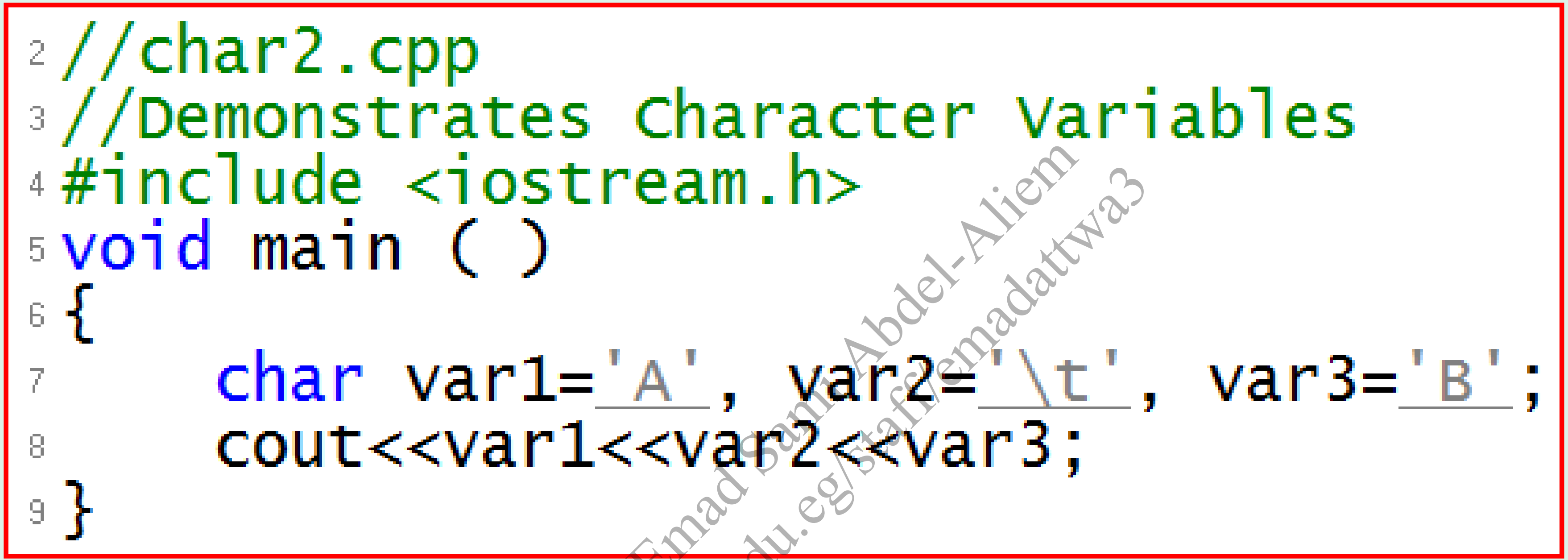


```
"D:\Faculty\2016-2017\2016-2017 _ Second Te
A       B
```

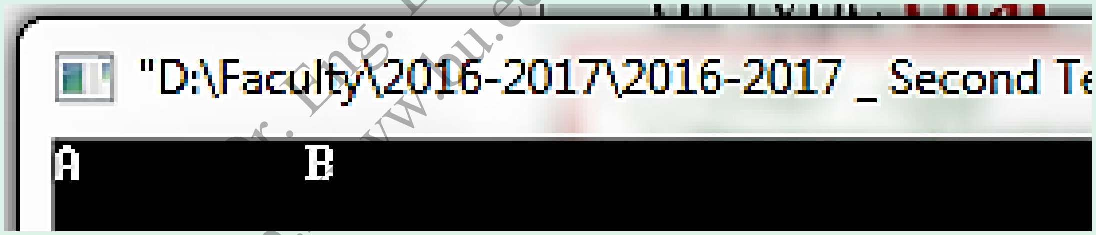
- To *assign* a value for **char** variable, you must use (‘ ’).
- You can *define* and *assign* more than one variable in the same statement.
- (**\t**) means: the program leaves a tab space on the screen.

2-5 Character Variable (char) …

- The previous example in a simple form.



```
//char2.cpp
//Demonstrates Character Variables
#include <iostream.h>
void main ( )
{
      char var1='A', var2='\t', var3='B';
      cout<<var1<<var2<<var3;
}
```

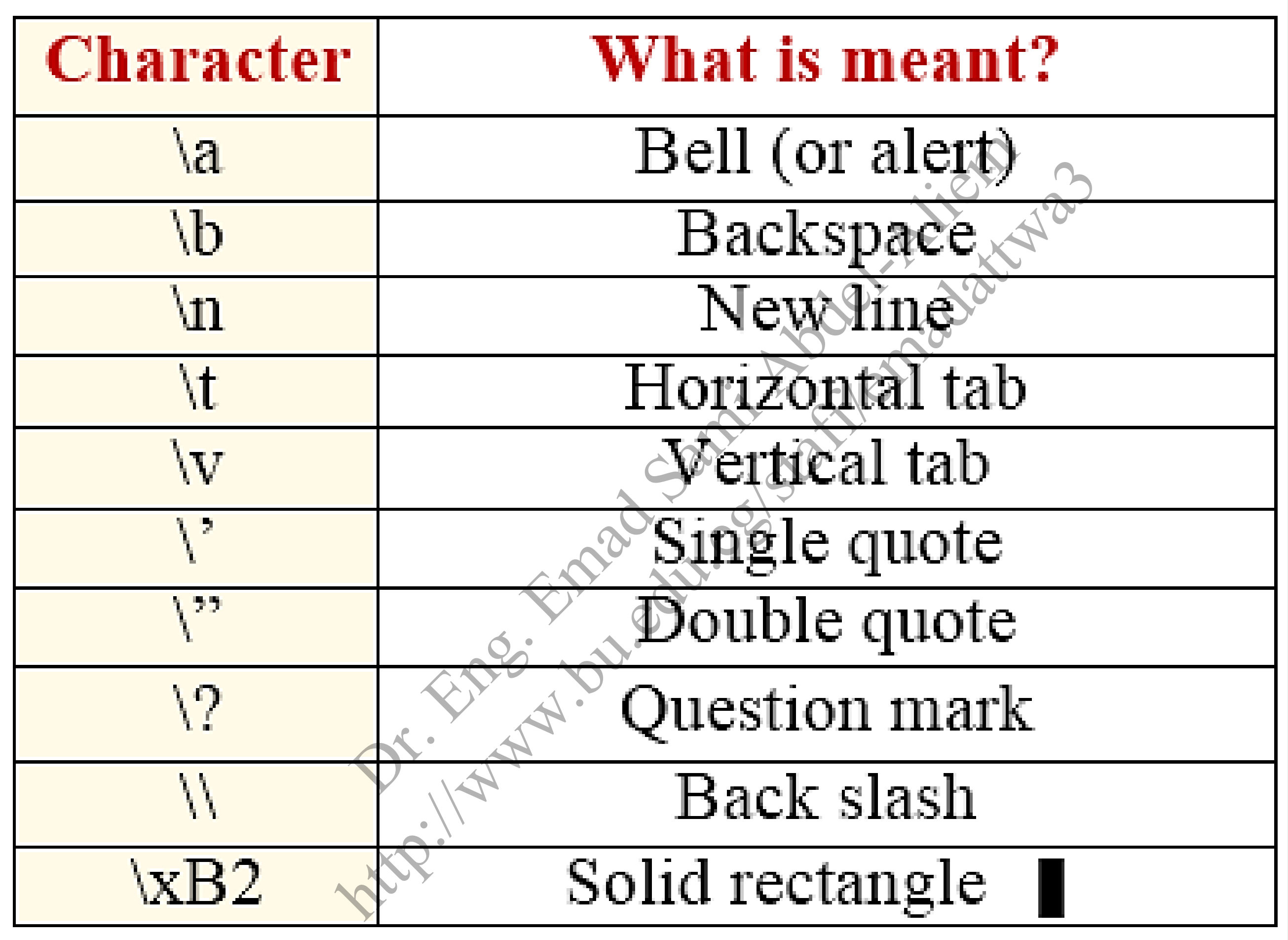


"D:\Faculty\2016-2017\2016-2017 _ Second Te

```
A       B
```

2-5 Character Variable (char) …

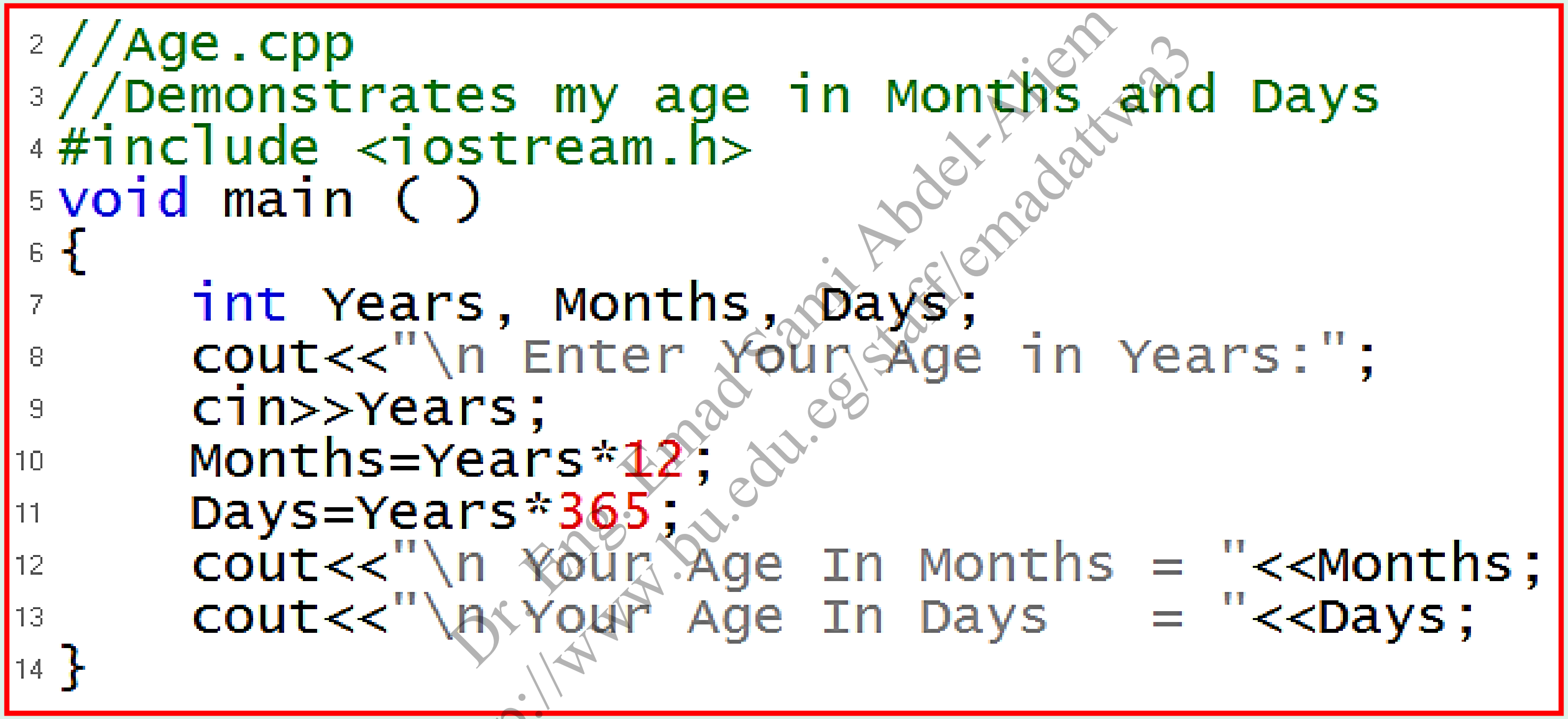
- The table shows special characters and its meaning.



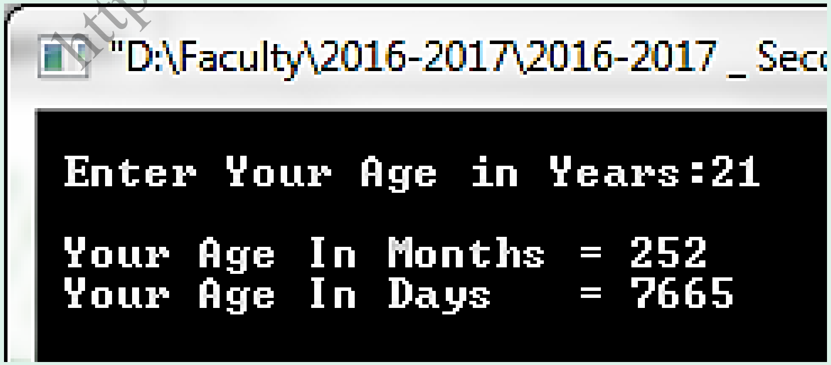
Character	What is meant?
\a	Bell (or alert)
\b	Backspace
\n	New line
\t	Horizontal tab
\v	Vertical tab
\’	Single quote
\”	Double quote
\?	Question mark
\\	Back slash
\xB2	Solid rectangle █

Quiz:

- Write a C++ program to convert your age in form of months and also in the form of days (when you enter your age in the program in years).



```cpp
//Age.cpp
//Demonstrates my age in Months and Days
#include <iostream.h>
void main ( )
{
       int Years, Months, Days;
       cout<<"\n Enter Your Age in Years:";
       cin>>Years;
       Months=Years*12;
       Days=Years*365;
       cout<<"\n Your Age In Months = "<<Months;
       cout<<"\n Your Age In Days   = "<<Days;
}
```



"D:\Faculty\2016-2017\2016-2017 _ Seco

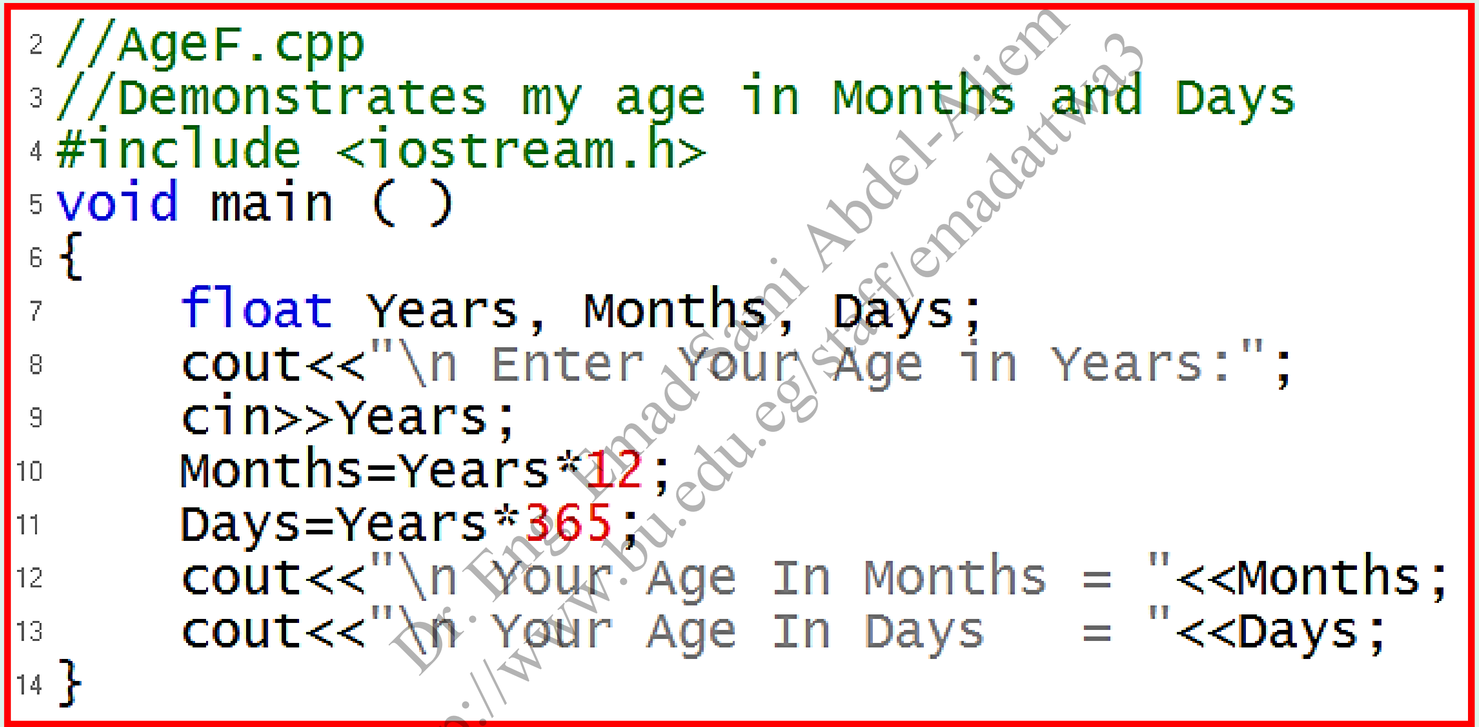
```
 Enter Your Age in Years:21

 Your Age In Months  = 252
 Your Age In Days    = 7665
```

2-6 Float Variable (**float**)

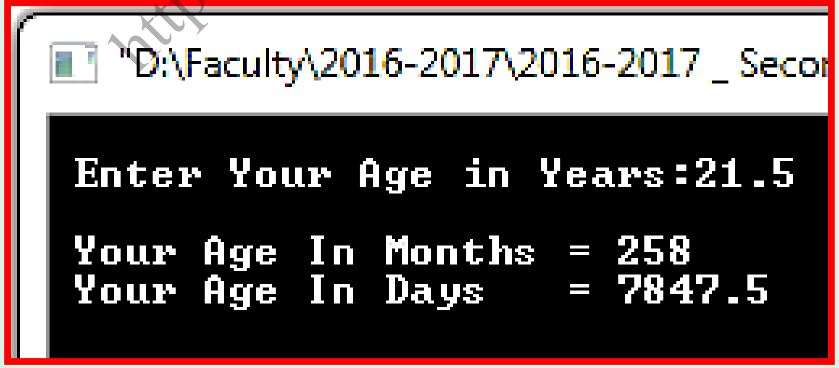
- We talked about the variables of type **int** and **char**. Both of them represent numbers without a fractional part.
- **Floating point** variables is the variable that can be represented using numbers that have decimal point, e.g., 1.50, 1.35, 2.367, …etc.
- **Floating point** variables represent also the real numbers which are used for measurable quantities like: distance, area and weight.
- **Floating point** variable which can carry decimal numbers also it can carry **int** numbers (e.g., the integer number 12 is considered as a float number of 12.0).

2-6 Float Variable (float) …

- Write a C++ program to convert your age in form of months and also in the form of days (when you enter your age in the program in years). [use float numbers]. AgeF.cpp



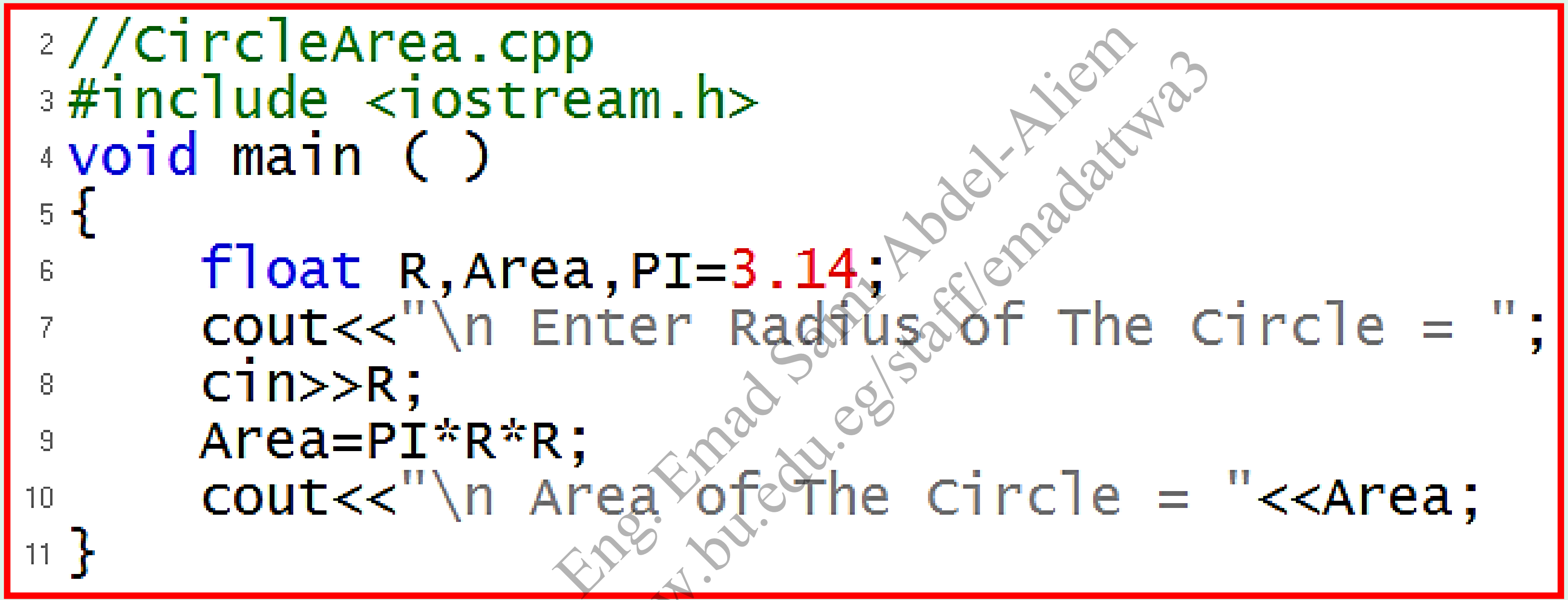
```
//AgeF.cpp
//Demonstrates my age in Months and Days
#include <iostream.h>
void main ( )
{
      float Years, Months, Days;
      cout<<"\n Enter Your Age in Years:";
      cin>>Years;
      Months=Years*12;
      Days=Years*365;
      cout<<"\n Your Age In Months = "<<Months;
      cout<<"\n Your Age In Days    = "<<Days;
}
```



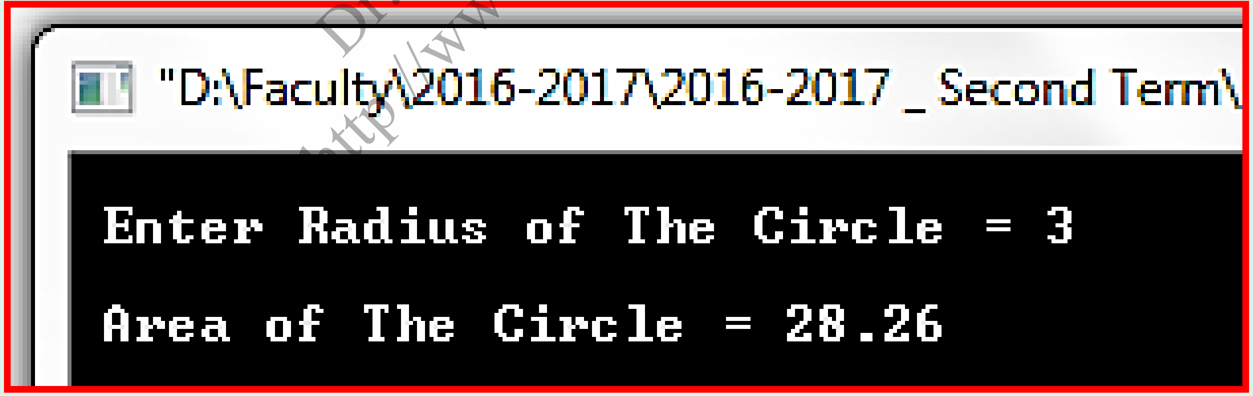
```
Enter Your Age in Years:21.5

Your Age In Months  = 258
Your Age In Days    = 7847.5
```

2-6 Float Variable (float) …

- Write a C++ program to calculate area of a circle. (the user enter the radius of the circle) CircleArea.cpp



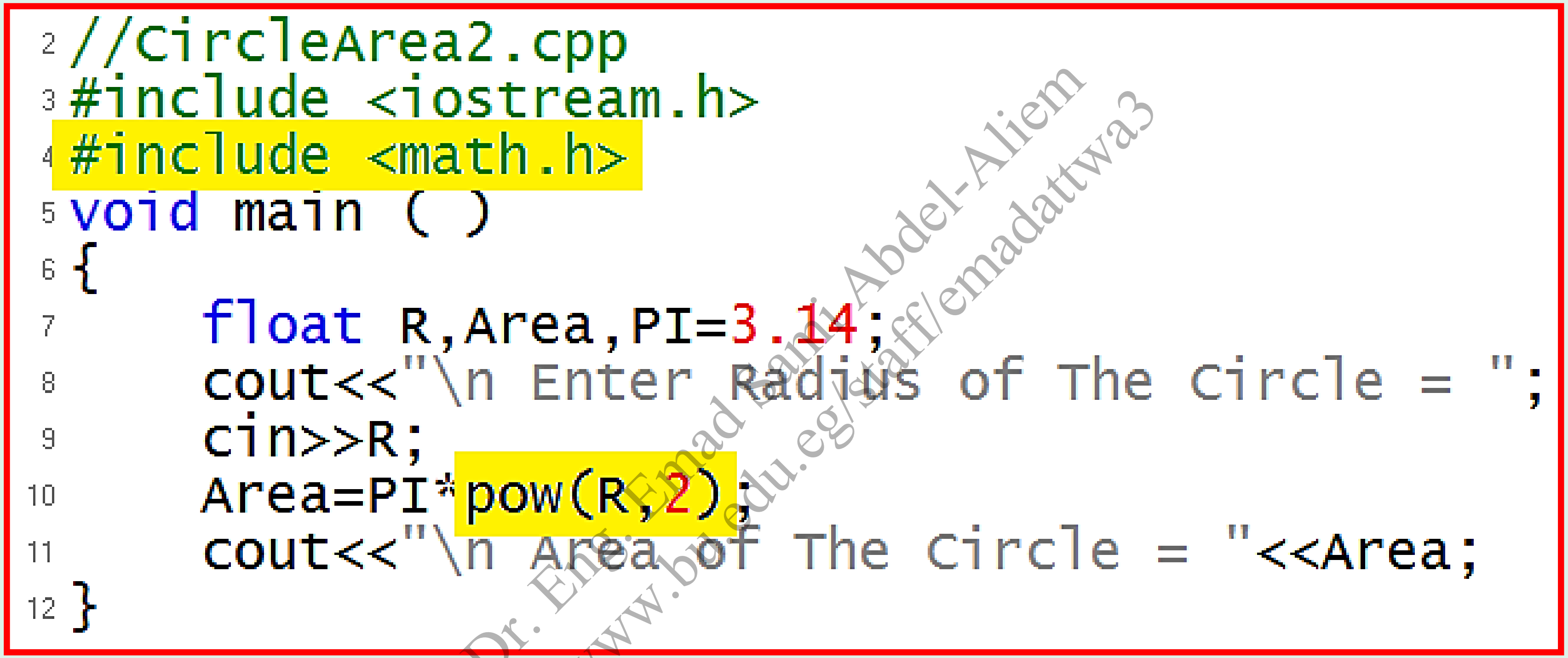
```
//CircleArea.cpp
#include <iostream.h>
void main ( )
{
	float R,Area,PI=3.14;
	cout<<"\n Enter Radius of The Circle = ";
	cin>>R;
	Area=PI*R*R;
	cout<<"\n Area of The Circle = "<<Area;
}
```



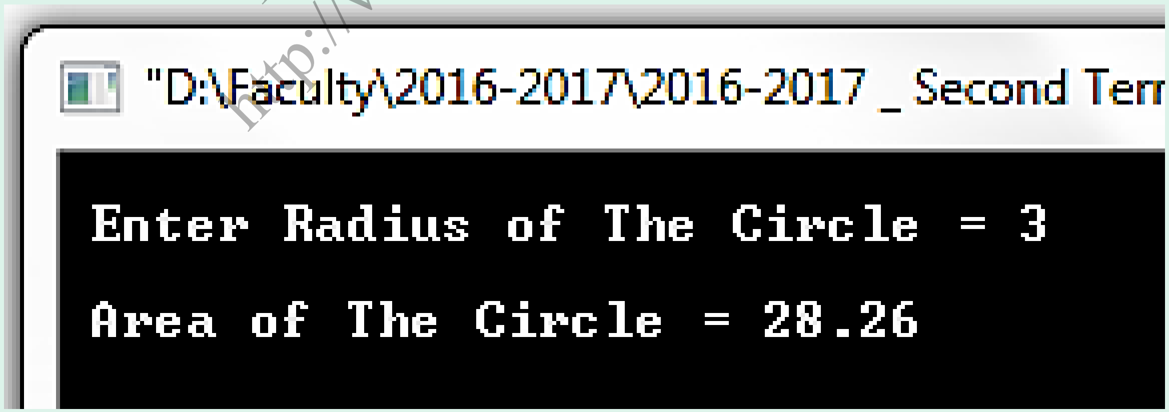
```
Enter Radius of The Circle = 3

Area of The Circle = 28.26
```

2-6 Float Variable (float) …

- Write a C++ program to calculate area of a circle. (the user enter the radius). لاحظ الفرق من البرنامج السابق CircleArea2.cpp



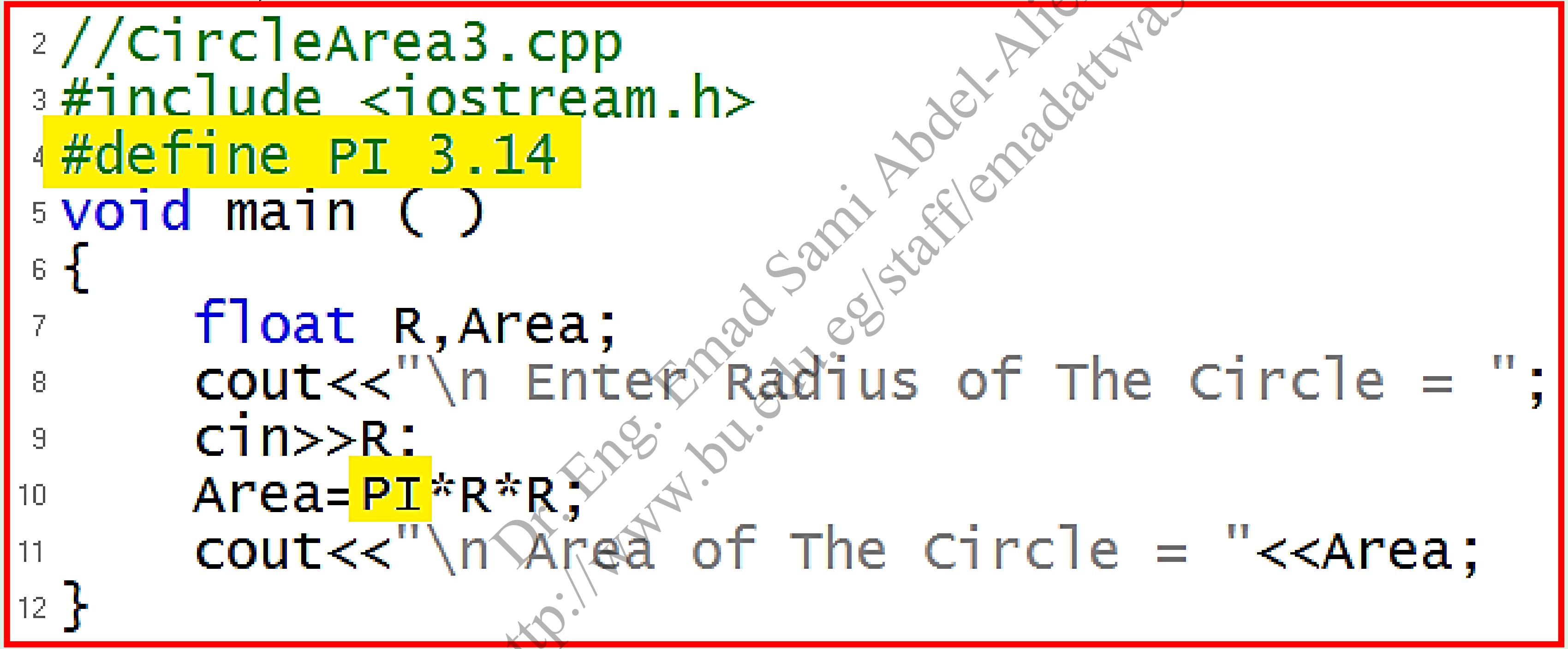
```
//CircleArea2.cpp
#include <iostream.h>
#include <math.h>
void main ( )
{
     float R,Area,PI=3.14;
     cout<<"\n Enter Radius of The Circle = ";
     cin>>R;
     Area=PI*pow(R,2);
     cout<<"\n Area of The Circle = "<<Area;
}
```



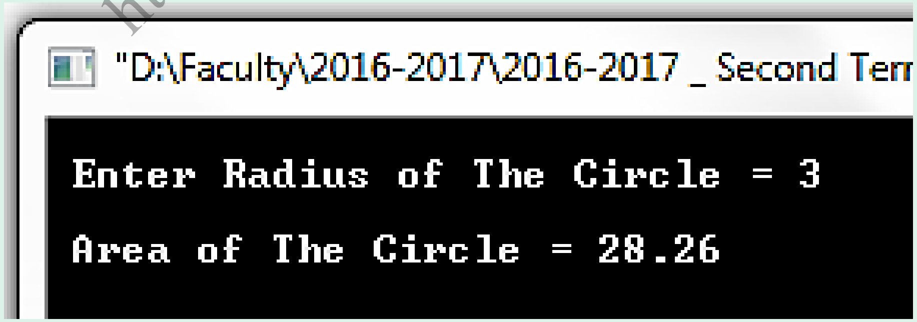
```
Enter Radius of The Circle = 3

Area of The Circle = 28.26
```

2-6 Float Variable (float) …

- يمكن تعريف الثابت لحطة إستخدامه كما فى المثالين السابقين أو(بإستخدام طريقة التعرف define)
- Write a C++ program to calculate area of a circle. (the user enter the radius). لاحظ الفرق عن البرنامجين السابقين CircleArea3.cpp



```
//CircleArea3.cpp
#include <iostream.h>
#define PI 3.14
void main ( )
{
	float R,Area;
	cout<<"\n Enter Radius of The Circle = ";
	cin>>R;
	Area=PI*R*R;
	cout<<"\n Area of The Circle = "<<Area;
}
```

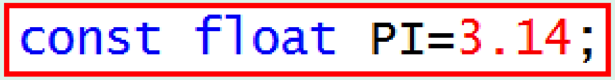


```
Enter Radius of The Circle = 3

Area of The Circle = 28.26
```

2-7 Naming constants and other Notes

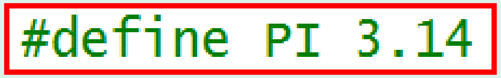
- ***Note 1:*** There are two methods to define the constants:

Method (1):



```
const float PI=3.14;
```

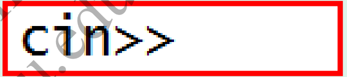
Method (2):



```
#define PI 3.14
```

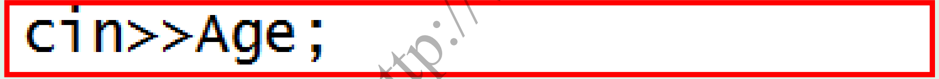
The Method (1) in defining the constant is better than Method (2) because you can determine in this statement the *variable type* of the constant number (the variable type may be: **int**, **float**, **double**, …etc). Also, in this method this value of PI is of type **float** and it can not be changed later in the program. If the programmer mistakenly tries to change it, the compiler will complain about it.

- ***Note 2***: input manipulator

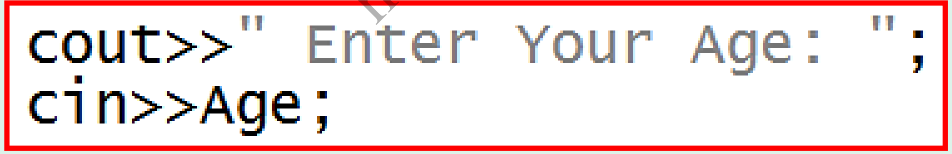


```
cin>>
```

You should print a certain message before using the manipulator cin to point for the user what he must input. Look at the difference:

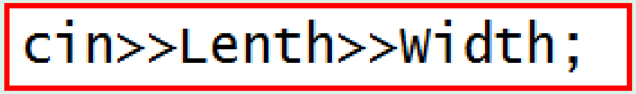


```
cin>>Age;
```



```
cout>>" Enter Your Age: ";
cin>>Age;
```

You can enter more than one input in the same cin statement:

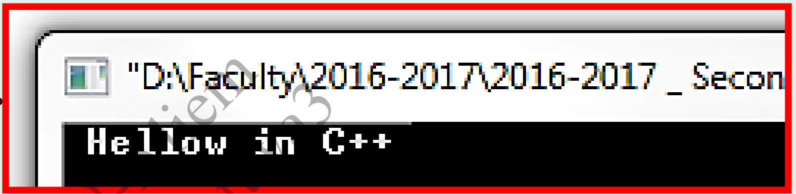


```
cin>>Lenth>>Width;
```

2-7 Naming constants and other Notes …

- ***Note 3:*** The manipulator used for print the output `cout<<` is used to print a certain message on the screen.

```
#include <iostream.h>
void main ( )
{
  cout<<" Hellow in C++ ";
}
```

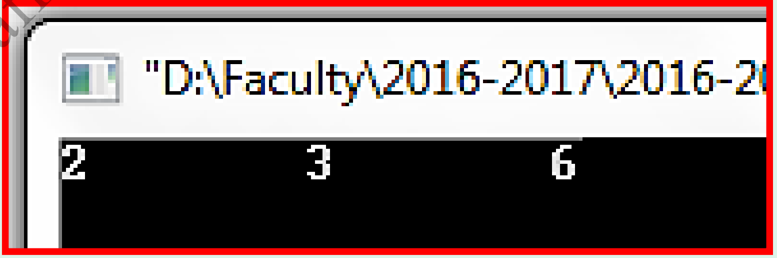


"D:\Faculty\2016-2017\2016-2017 _ Secon

```
 Hellow in C++
```

- ***Note 4***: Note the difference in the following output:

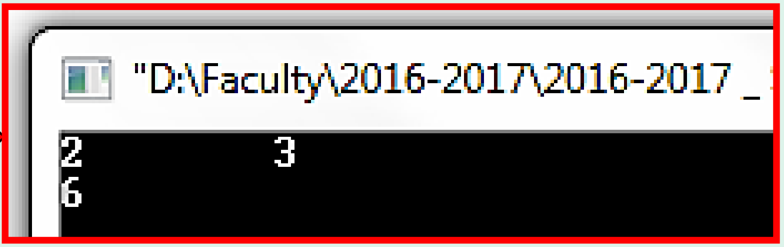
```
#include <iostream.h>
void main ( )
{
  float A,L=2,W=3;
  A=L*W;
  cout<<L<<"\t"<<W<<"\t"<<A;
}
```



"D:\Faculty\2016-2017\2016-2

```
2       3       6
```

```
#include <iostream.h>
void main ( )
{
  float A,L=2,W=3;
  A=L*W;
  cout<<L<<"\t"<<W<<"\n"<<A;
}
```



"D:\Faculty\2016-2017\2016-2017 _

```
2       3
6
```

- ***Note 5***: the manipulator (\n) means a new line. The manipulator (endl) means ends the line and start from a new line.

2-8 Variable Type (long int)

- **long int** variable type can hold integers in the range:
  -2,147,483,684 to 2,147,483,684.
- The variable type **short int** is too small to be used for storing the values.
- Memory size for **long int** is 4 bytes.
- Memory size for **short int** is 2 bytes.
- *What is the preferred time for using unsigned integers?*

Answer: we use **unsigned integer** for variables which are always positive. For example: number of students in a class, length of a room, … etc. In these cases there is no possibility for using negative values. Therefore, it is preferred to use **unsigned integer** to get maximum benefit of your memory. The variable type **short int** takes values (-32,768 to 32,768) and occupies 2 bytes of memory. The variable type **unsigned short int** takes values (0 to 65,535) and also occupies 2 bytes of memory.

2-9 Table of Variable Type and Its Size

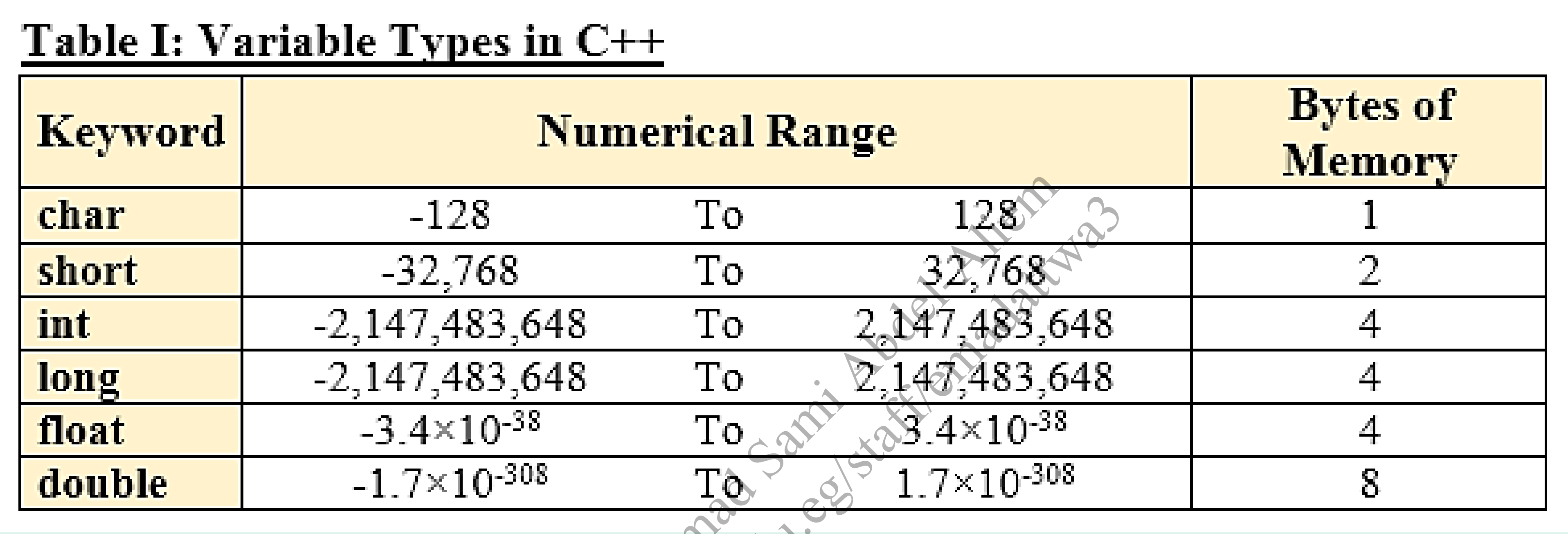
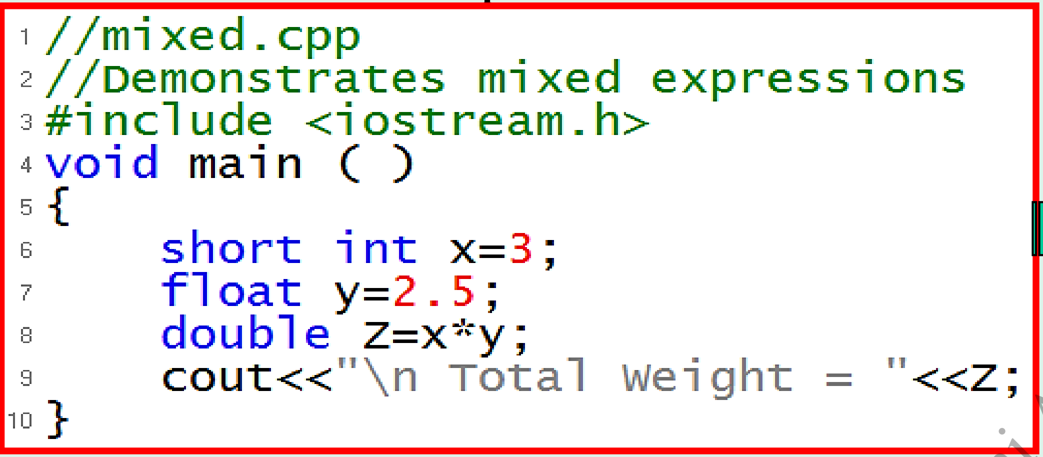


Table I: Variable Types in C++

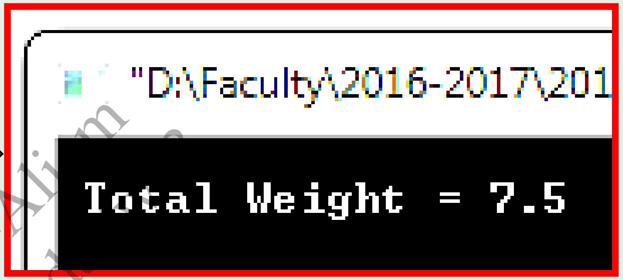
Keyword	Numerical Range			Bytes of Memory
char	-128	To	128	1
short	-32,768	To	32,768	2
int	-2,147,483,648	To	2,147,483,648	4
long	-2,147,483,648	To	2,147,483,648	4
float	-3.4×10^{-38}	To	3.4×10^{-38}	4
double	-1.7×10^{-308}	To	1.7×10^{-308}	8

2-10 Automatic Data Conversion

- Here an example shows automatic data conversion: mixed.cpp



```
//mixed.cpp
//Demonstrates mixed expressions
#include <iostream.h>
void main ( )
{
      short int x=3;
      float y=2.5;
      double Z=x*y;
      cout<<"\n Total Weight = "<<Z;
}
```



```
"D:\Faculty\2016-2017\201
Total Weight = 7.5
```

- A variable of type **short int** (x) is multiplied by a variable of type **float** (y) and the result is assigned to a variable of type **double** (z). This is a mixed expression, the compiler solves this problem by automatically converting lower order data type to higher order ones as in the following table II [i.e., the **short int** value of (x) is converted to type **float** and stored in temporary variable before being multiplied by the **float** (y). The result (still of type **float**) is then converted to **double** so that it can be assigned to the **double** variable (z)].
- Never mind, these conversions take place automatically by C++.

2-10 Automatic Data Conversion …

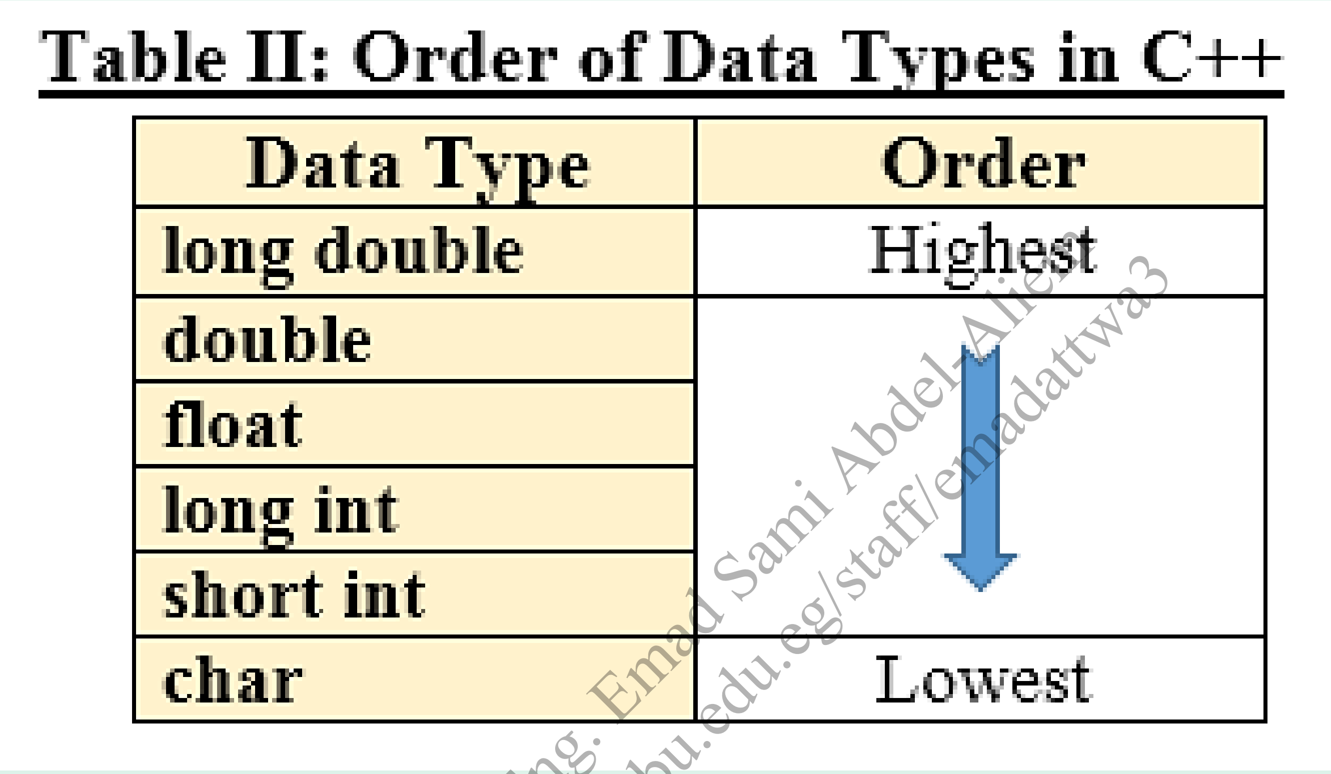


Table II: Order of Data Types in C++

Data Type	Order
long double	Highest
double	↓
float	
long int	
short int	
char	Lowest

2-11 Assignment (2)