University of Sallahadin
College of Engineering
Electrical Engineering Dept.

Computer Programming / C++ Programming Language

Chapter -2 (Basics of C++ language)

Ahmed Khalid Ahmed

Email address : aeng@engineer.com

Website : www.computer-lab.us

Chapter -2 (Basics of C++ language)

2.1 The Input/ Output Command.

2.2 The Variables.

2.3 Data Values.

2.3.1 Integer Values.

2.3.2 Floating-Point Values (Real Numbers).

2.3.3 Character Value.

2.4 Declaration Of Variables.

2.5 Displaying a Variable's Address.

2.6 Initialization of Variables.

2.7 Scope of Variables.

2.8 Constants.

2.9 Defined Constants (#define).

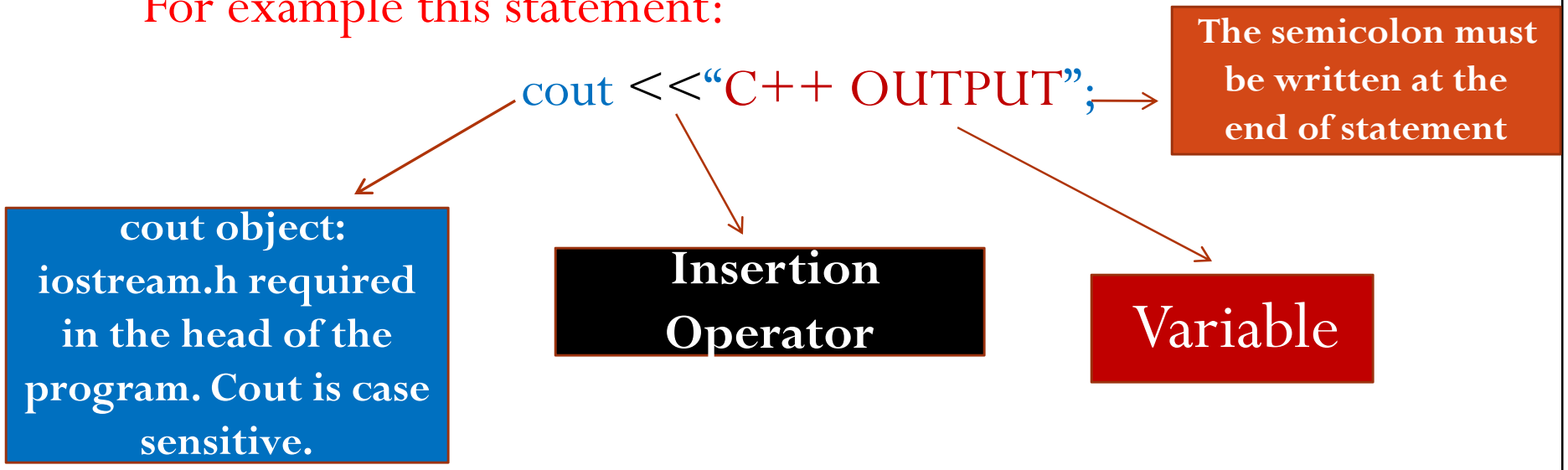
2.10 Declared Constants (const).

2.1 The Input/ Output Command.

- In the iostream C++ library two data streams supported (cin and cout) for standard input and output operations.
- Cout is derived from Console out (Output Unit).

How to use it?

For example this statement:



2.1 The Input/ Output Command (cont.)

Try it by your self:

```
#include <iostream.h>
void main()
{
cout<<"This is My first C++ Program";
}
```

Note that the Output will be:

This is My first C++ Program

You will see the above text in the black screen when you run the program….

2.1 The Input/ Output Command (cont.)

Lab. Work!

Try to print out your full name using the computer in computer laboratory?

Hint: use the same example shown in previous slide.

- Note that the cout is responsible in printing out any string or any thing behind the insertion operator(<<), also cout represent the black output screen.
- The cin object is an object that causes the program to wait (after executing) for the user to type the required data by keyboard.

2.1 The Input/ Output Command (cont.)

How to use it?

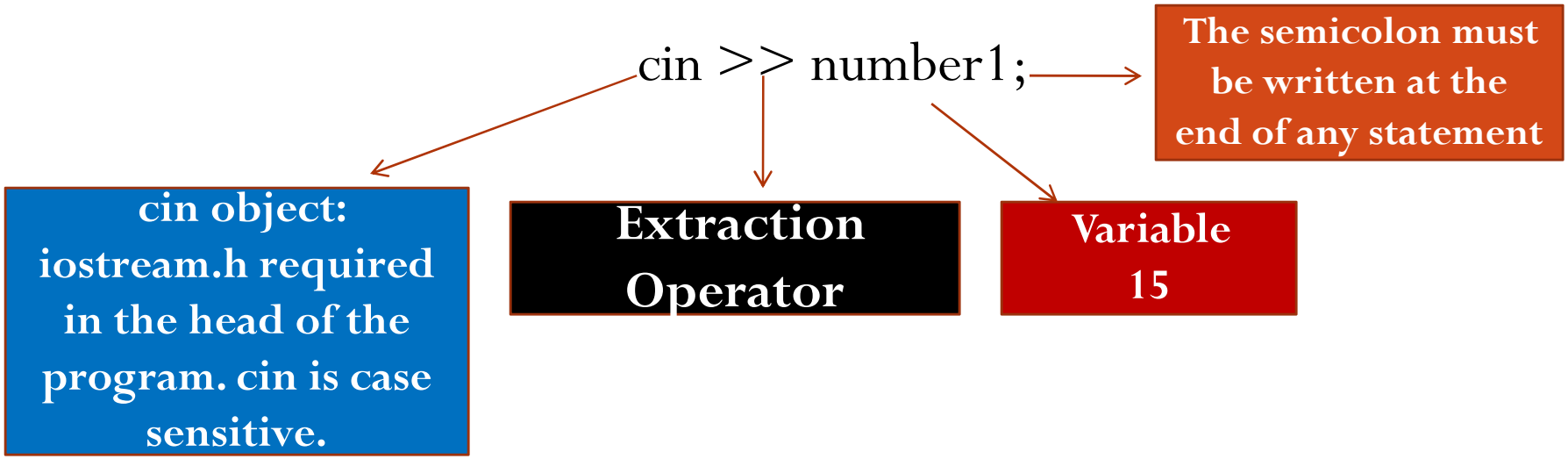
For example this statement:

```
cin>>number1;
```

When the statement executed the program will wait for the user to enter a number by keyboard, and the number value is placed in the variable named number1.

number1 is a variable, and it can take any value when the program is executed.

2.1 The Input/ Output Command (cont.)



Try out this!

```
#include <iostream.h>
void main()
{
int number1;
cin>>number1;
}
```

2.1 The Input/ Output Command (cont.)

- cin is representing the keyboard, also whenever cin is used it means that you have to enter something in the black output screen after executing the program.

2.2 The Variables.

• All integers, real numbers and other values used in a computer program are stored in the computer's memory.

Put a 45 in location 1624
Put a 12 in location 1625

The example above is the instructions used in low level languages to store numbers inside memory locations

2.2 The Variables (cont).

- In high level languages symbolic names are used in place of actual memory address, these symbolic names called variables.
- The term variable used because the value stored in the variable can change or vary.
- In C++ the variable is a name given by the programmer that is used to refer to computer location memory.
- The variable names has rules that the programmer should follow:

1. The variable name must begin with letter or underscore (_) and may contain only letters, underscore, or digits. It cannot contain any blanks, commas, or special symbols such as: () , & ,$, @, . \ , ?.

2.2 The Variables (cont).

2. A variable name cannot be a keyword such as:

auto, const ,double ,float ,int ,short , struct , unsigned break , continue, else, for, long ,signed ,switch, void, case default, enum, goto, register, sizeof, typedef, volatile, char, do, extern, if, return, static, union, while, asm dynamic_cast, namespace, reinterpret_cast, try, bool, explicit, new, static_cast, typeid, catch, false, operator, template, typename, class, friend, private, this, using, const_cast, inline, public, throw, virtual, delete, mutable, protected, true, wchar_t.

3. The variable name cannot consist of more than 31 characters.

2.2 The Variables (cont).

- In C++ language to store 45 in memory location 1624 and 12 in 1625, then to add the result of addition of 45 and 12 into location 1622 the following statements should be written:

  Num1 = 45;

  Num2 = 12;

  Total = Num1+Num2;

- C++ is a Case-Sensitive language…

  Total is not equal to total

  Num1 is not equal to num1 …….etc

2.3 Data Values.

2.3.1 Integer Values.

- An integer value is zero or any positive or negative number without a decimal point.
- Examples of valid integer (int) values are: 0 , 5, -10 , +25, 1000, -2345, +36.
- Examples of invalid integer values are: $234, 3,44 , 3. .

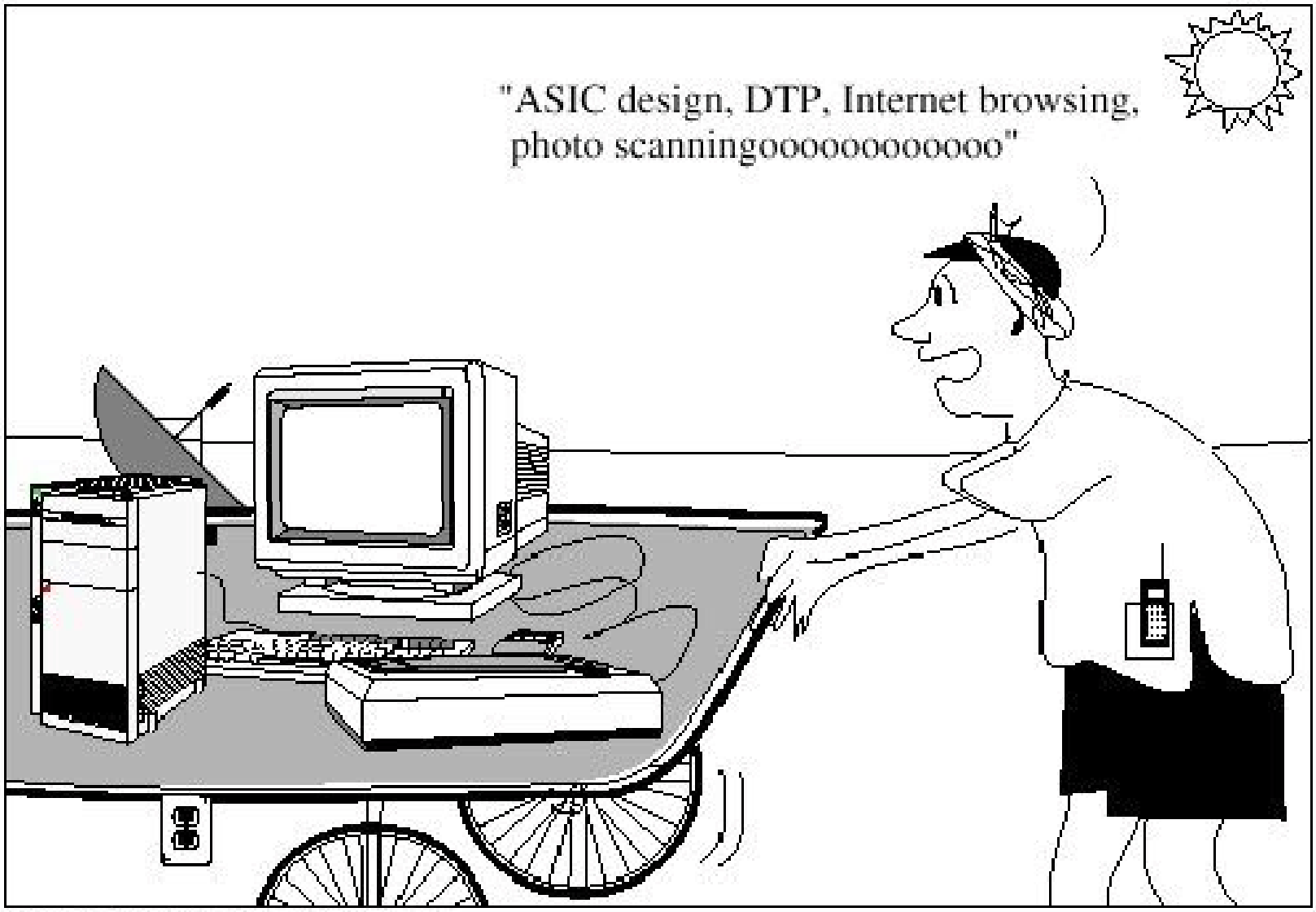
2.3.2 Floating point numbers or (real numbers) are numbers with decimal point (signed and unsigned) for example: +10.3, 5. , 6.9 , 0.0 , 0.3, +2.

- Invalid Floating numbers are : 3,333.9 , 24, 1,23, $88.0.

2.3.3 Character Value: character includes the letters of alphabet (both upper and lower case), the ten digits (0) to (9) and special symbols (+,$,#, -, !) and so on.

- A single character value is any one letter, digit, or special symbol enclosed by single quotes, examples of valid character values are: "a", "A" , "$", "7" and etc. .
- To show a string of more than one character the enclosed with two quotes for example : "This is the string of letters"

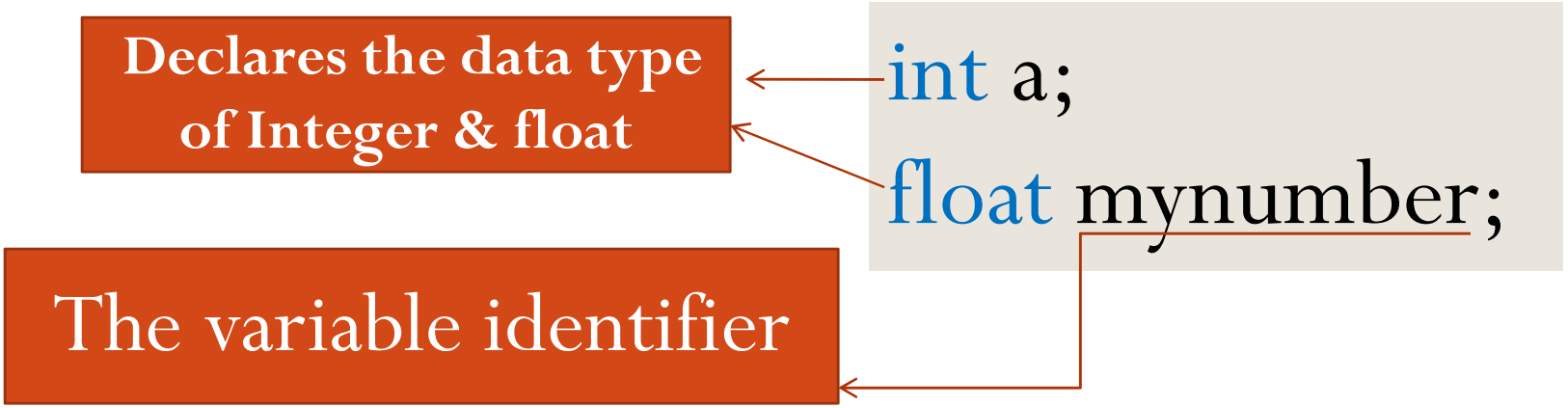
Name	Description	Size*	Range*
`char`	Character or small integer.	1byte	signed: -128 to 127 unsigned: 0 to 255
`short int(short)`	Short Integer.	2bytes	signed: -32768 to 32767 unsigned: 0 to 65535
`int`	Integer.	4bytes	signed: -2147483648 to 2147483647 unsigned: 0 to 4294967295
`long int (long)`	Long integer.	4bytes	signed: -2147483648 to 2147483647 unsigned: 0 to 4294967295
`bool`	Boolean value. It can take one of two values: true or false.	1byte	`true` or `false`
`float`	Floating point number.	4bytes	+/- 3.4e +/- 38 (~7 digits)
`double`	Double precision floating point number.	8bytes	+/- 1.7e +/- 308 (~15 digits)
`long double`	Long double precision floating point number.	8bytes	+/- 1.7e +/- 308 (~15 digits)



Janardhana Swamy & Raju (swamyvc@hotmail.com)

2.4 Declaration Of Variables.

- In order to use variables in C++ Programming Language, the variable must be declared which of the data types wanted to be.
- The syntax to declare a variable is to write the data type that wanted for example (int , float, short….etc.) followed by a valid variable identifier (the name of variable) for example:



2.4 Declaration Of Variables.(cont.)

```
int a;
int b;
int c;
int a, b, c;
unsigned short number1;
signed int my_variable;
unsigned mynumber;
double x, y, z;
char name;
```

2.5 Displaying a Variable's Address.

- Every variable has three major items associated with it. Its data type (int, float, double,char, ….etc.), the actual value stored inside the variable (24, 23.9, "abc"), and the address of the variable.
- How many locations are actually used for the variable depends on the variables data type.
- To determine the location of any variable in C++ programming language, the C++'s address operator (&) is used before the variable name.

2.5 Displaying a Variable's Address.

- For example to find the location of a variable named (Myvariable) we will write the following statement:

```
cout<<&Myvariable<<endl;
```

Example :

```
#include <iostream.h>
void main()
{
int My_data;
cout<<"The location of  My_data is"<<" "<<&My_data<<endl;
}
```

The output will be:

The location of My_data is 0x0012FF44

2.6 Initialization of Variables.

- After declaration of the variable, the variable should be initialized by giving a value according to the type of the variable, for example:

```
int Data;    // declaration of the variable (Data)
Data = 4;  // initialization of the variable (Data)
```

Note that we give the variable a value (4) which it is integer value, it is not from the programmers habit to declare a variable with int then initialize it with a number of another data type for example 4.6, this will cause data loss. If the programmer deliberately did it! It is another story.

2.6 Initialization of Variables.(cont.)

- In initializing variables, the variable name must be at the left of the "=" sign, and the value at the right of the "=". For example data = 5;

2.7 Scope Of Variables.

In C++ programming language the variable can be declared anywhere in the source file….

2.7 Scope Of Variables. (cont.)

```
#include <iostream.h>
int integer;
char character;
unsigned int Number;
void main()
{
unsigned short Age;
float Anumber, anothernumber;
cout<<"enter your age"<<endl;
cin>>character;
}
```

Global variable: these declared variables are called global variables because it can be used anywhere in the source file, so it can be used inside main function and other functions inside source file. There is no need to declare global variables again inside the functions.

Local Variables: these variables inside the main function is called local variables because it is used only inside main function only and can not be used out side the main function except if we declare in again.

2.8 Constants.

- In addition to decimal numbers C++ supports Hexadecimal base (16) and Octal base (8) numbers.

Examples of numbers in C++

Decimal	75
Octal	0113
Hexa	0x4b

2.9 Defined Constants (#define).

- We can define our own names for constants that might we use it inside our codes in C++ programming language by using #define preprocessor directive.
- But how to use it? The answer is using the syntax below:

#define identifier value

For example

```
#define PI 3.14159265
#define newline "\n"
#define width 100
```

2.9 Defined Constants (#define). (cont.)

- We can use PI in the main function directly without declaring it inside the main function.

Example : Write a C++ program to evaluate the following equation:

$$Y = 2\pi.X$$

Use #define to define π and the value of X may be any value entered to your program.

2.9 Defined Constants (#define). (cont.)

Solution:

```
#include <iostream.h>
#define PI 3.141565
void main()
{
    double Y;
    float X;
    cout<<"Please Enter the Value of X:"<<" ";
    cin>>X;
    Y=2*PI*X;
    cout<<"The result of equation :(Y=2*PI*X)is"<<" "<<Y<<endl;
}
```

2.10 Declared Constants (const).

- With the const we can declare constants with specific type exactly as we did with variables:

Example:

```
const  int width = 100;
const  char  tab= "\t";
const zip = 12440;
```

Hint: To write equations in C++! For example if we have the following equation: (Y=X.Y/2+π) we will write Y=(X*Y)/(2+ π);

With out the brackets the answer will be wrong.

Lab. Work

1- Rewrite all examples illustrated in Chapter Two.

2- Write a C++ Program to evaluate the below equation:

$$Y = X^3 \times Z$$