



**Computer Programming (a)**

**E1123**

**Lecture 2**

**C++ Basics**



**INSTRUCTOR**

**DR / AYMAN SOLIMAN**

## ➤ Contents

- 1) Course Information
- 2) Objectives
- 3) C++ compiler directives
- 4) Libraries and the C++ Standard Library
- 5) First Program
- 6) Second Program

# 1) Course Information.

**Lectures:** Tuesday, (10:40 - 11:25 AM)

**Office Hours:** Tuesday, Thursday.

**Prerequisite:** E1021 - E1022

**References:**

- **C++ Programming: From Problem Analysis to Program Design, Fifth Edition D.S. Malik**
- **Object-Oriented Programming Using C++, Fourth Edition Joyce Farrell**
- **[www.learncpp.com](http://www.learncpp.com)**

**Instructor:**

**Dr. Ayman Soliman**

**[Ayman.mohamed01@bhit.bu.edu.eg](mailto:Ayman.mohamed01@bhit.bu.edu.eg)**

**TAs:**

**Eng. Nada Elmeligy**  
**Eng. Enas Mohamed**  
**Eng. Rehab Ibrahim**  
**Eng. Mohamed Mostafa**

**Eng. Ahmed Ragab**  
**Eng. Mohamed Abd Satar**  
**Eng. Mahmoud Osama**  
**Eng. Khalid Diao**

## 2) Objectives

- To be able to write simple computer programs in C.
- To be able to use simple input and output statements.
- To become familiar with fundamental data types.
- To understand computer memory concepts.
- To be able to use arithmetic operators.
- To understand the precedence of arithmetic operators.



### 3) C++ compiler directives

- Compiler directives appear in green color in C++.
- The **#include** directive tells the compiler to include some already existing C++ code in your program.
- The included file is then linked with the program.
- There are two forms of **#include** statements:

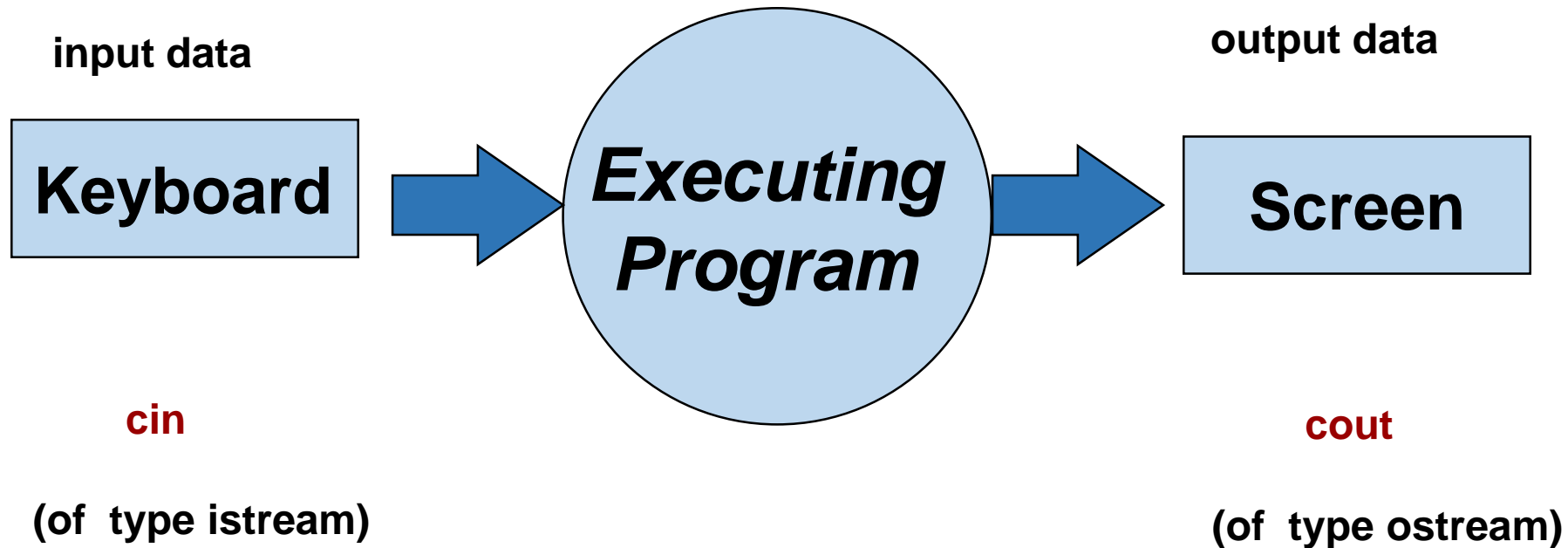
**#include <iostream> //for pre-defined files**

- the C++ label for a standard header file for input and output streams

**#include "my\_lib.h" //for user-defined files**

# Keyboard and Screen, I/O

```
#include <iostream>
```



# Input

- Variable **cin** is predefined to denote an **input stream from the standard input device** (the keyboard)
- The extraction operator **>>** called “**get from**”. The left operand is a **stream expression**, such as **cin**--the right operand is a variable of simple type.
- Operator **>>** attempts to **extract** the next item from the input stream **and store** its value in the right operand variable.

```
cin >> Variable1 >> Variable2 ...;
```

# Output

- To do input/output, at the beginning of your program you must insert `#include <iostream>`  
`using cout; using endl;`
- C++ uses streams for input and output
- *stream* - is a sequence of data to be read (*input stream*) or a sequence of data generated by the program to be output (*output stream*)
- Variable **cout** is predefined to denote an **output stream that goes to the standard output device** (display screen).
- The insertion operator `<<` called “put to”.
- The left operand is a stream expression, such as **cout**. The right operand is an **expression** of simple type or a **string constant**.



# Output Statements Styles

## Syntax

```
cout << Expression1 << Expression2 ...;
```

- cout statements can be linked together using << operator.
- These examples yield the same output:

```
cout << "The grades are " ;
```

```
cout << 90;
```

```
cout << "The grades are " << 90;
```

# How Extraction Operator works?

➤ Input is not entered until user presses **<ENTER>** key.

➤ Allows backspacing to correct.

➤ Skips whitespaces (space, tabs, etc.)

➤ Multiple inputs are stored in the order entered:

```
cin>>num1>>num2;
```

User inputs: **5 8**

Assigns **num1 = 5** and **num2 = 8**

➤ No difference between a single cin with multiple variables and multiple cin statements with one variable

```
cin>>num1>>num2;
```

```
cin>>num1;  
cin>>num2;
```

These examples yield the same output.

# Expressions

- An **expression** is a valid arrangement of variables, constants, and operators.
- In C++, each **expression** can be evaluated to compute a value of a given type
- In C++, an expression can be:
  - ❑ A variable or a constant (area, 22)
  - ❑ An operation ( $x + y$ ,  $z / 5$ )
  - ❑ Function call (calculaterectanglearea(5, 10))

# Comments

➤ Allow commentary to be included in program

➤ C++ has two conventions for comments

// single line comment (preferred)

/\* long comment \*/ (save for debugging)

➤ Typical uses

Identify program and who wrote it

Record when program was written

Add descriptions of modifications

# Escape sequences

- Escape sequences are used to represent certain special characters within [string literals](#) and [character literals](#).

Alert	\a	Makes an alert, such as a beep
Backspace	\b	Moves the cursor back one space
Formfeed	\f	Moves the cursor to next logical page
Newline	\n	Moves cursor to next line
Carriage return	\r	Moves cursor to beginning of line
Horizontal tab	\t	Prints a horizontal tab
Vertical tab	\v	Prints a vertical tab
Single quote	\'	Prints a single quote
Double quote	\"	Prints a double quote
Backslash	\\	Prints a backslash
Question mark	\?	Prints a question mark

# Preprocessor directives

The **preprocessor** is a separate program that runs just before the compiler when you compile your program. When you `#include` a file, the preprocessor copies the contents of the included file into the including file at the point of the `#include` directive.

**Directives** are specific instructions that start with a `#` symbol and end with a newline (NOT a semicolon).

There are two different types of directives

// The files or libraries that are part of the C++ standard library

```
#include <filename>
```

// You'll generally use this form for including your own header files

```
#include "filename.h"
```

## 4) Libraries and the C++ Standard Library

- A library is a collection of precompiled code (functions) that has been “packaged up” for reuse in many different programs such as **math** library, **sound** library and a **graphics** library.
- C++ comes with a library called the C++ standard library that provides additional functionality for your use, and it is divided into areas or libraries that provide a specific type of functionality.
- One of the most used parts of the C++ standard library is the iostream library, which contains functionality for writing to the screen (**cout**) and getting input (**cin**) from a console user.

## 5) First Program

**Preprocessor directives** tell the compiler to add the contents of the iostream header to the program that includes cout and cin.

```
#include <iostream>
```

This line is blank, and it is ignored by the compiler.

```
int main()  
{  
    cout << "Hello world!" ;  
    return 0;  
}
```

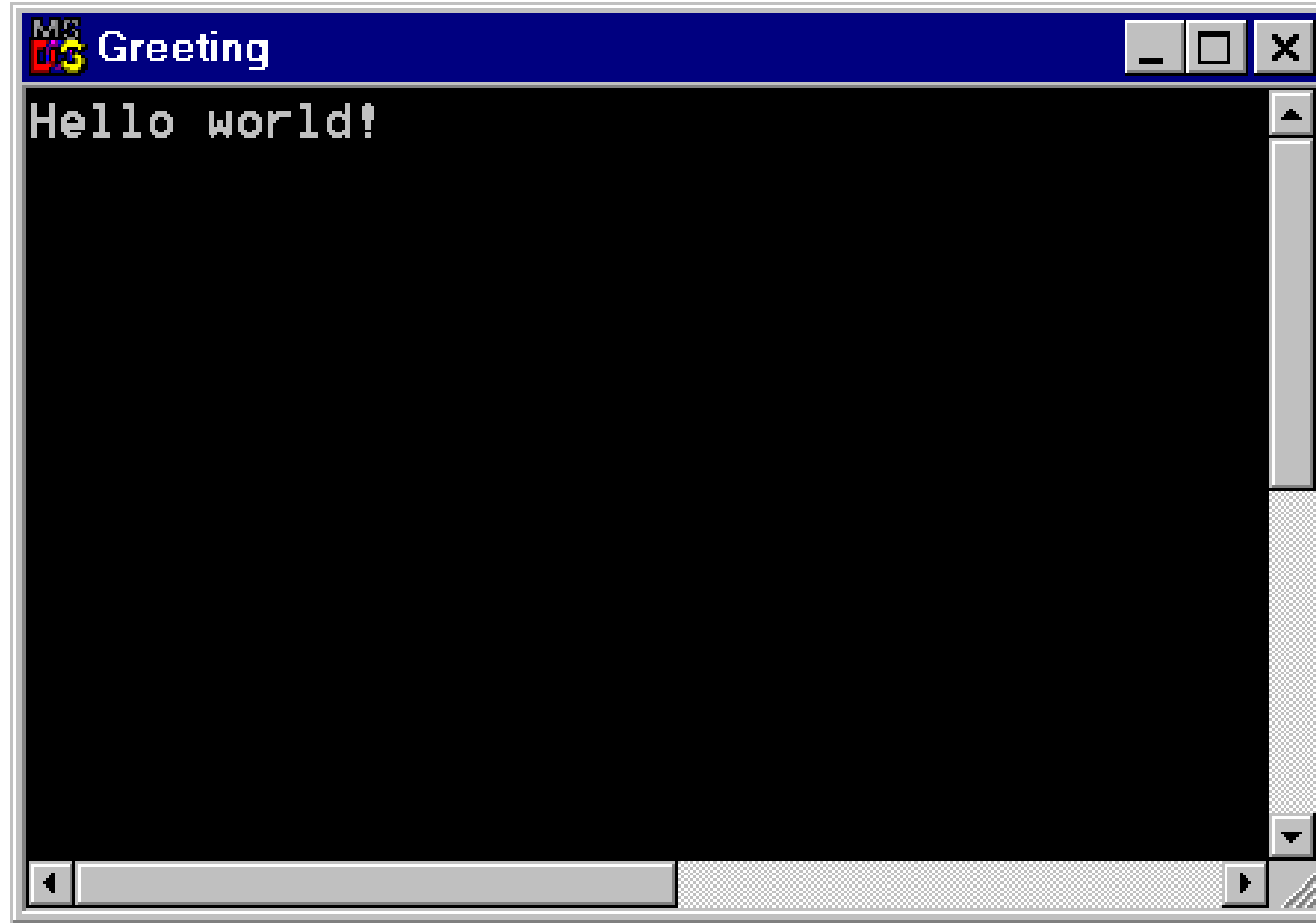
declaring the **main()** function, which is mandatory. Everything inside curly brace {} is a part of main() function.

The << symbol is an the **output operator**.

A **return statement** sends a value back to the operating system that indicates whether it was run successfully or not.



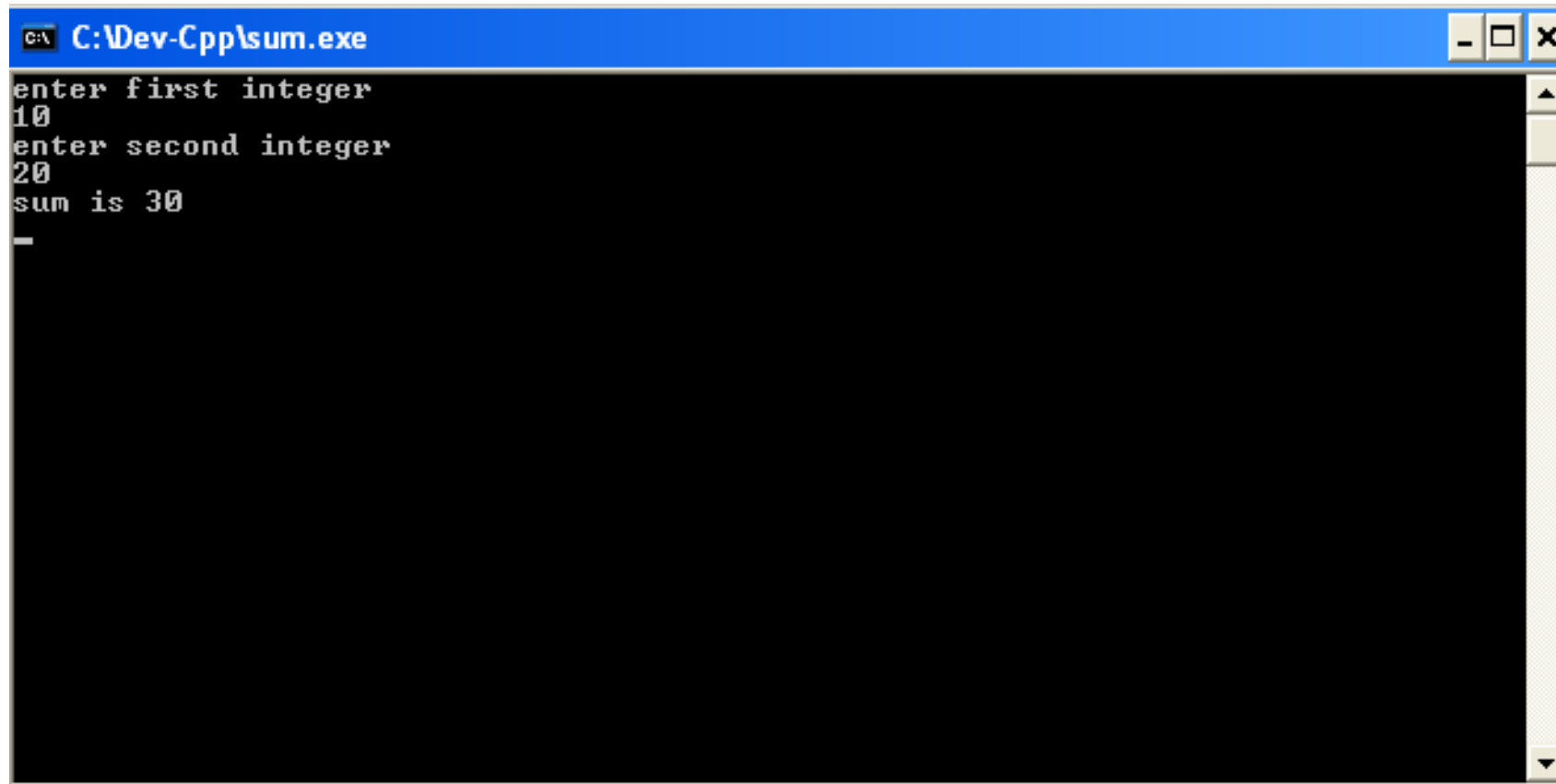
# Greeting Output



## 6) Second Program

```
1 //example
2 // program to add two numbers
3 #include <iostream.h>
4
5 int main()
6 {
7     int integer1, integer2, sum;           // declaration
8
9     cout << "Enter first integer\n";      // prompt
10    cin >> integer1;                       // read an integer
11    cout << "Enter second integer\n";     // prompt
12    cin >> integer2;                       // read an integer
13    sum = integer1 + integer2;            // assignment of sum
14    cout << "Sum is " << sum << endl;     // print sum
15
16    return 0;                             // indicate that program ended successfully
17 }
```

# Output



```
C:\Dev-Cpp\sum.exe
enter first integer
10
enter second integer
20
sum is 30
_
```

Thank  
you

