Introduction to C++ Programming Language

Dr. Sawsan M. Mahmoud
Computer Engineering
Faculty of Engineering
Al- Mustansiriyah University
2017-2018

Contact

- **Name:** Sawsan M. Mandalawi
- **Email:** sawsan.mandalawi@yahoo.com
- **Room:** Room 6 in the Department of Computer & Software Engineering
- **Lecture Time:** Tuesday 11:30 - 02:30
  Thursday 08:30 - 11:30
- **Notes:** Other office hours are available by an appointment. The contents of this syllabus is not to be changed during the current semester.

What is this Module all about?

- This course introduces the basic data structures used in computer software
  - Understand the data structures
  - Analyze the algorithms that use them
  - Know when to apply them

  Also, practice design and analysis of data structures and practice using these data structures by writing programs.
- On successful completion of this course, you will understand:
  - What the tools are for storing and processing common data types
  - Which tools are appropriate

  So that you will be able to make good design choices as a developer, project manager, or system customer

Module Content

- Introduction C++ programming language
- Declaration, Comments, Numbers, Variables, I/O data (cout, cin).
- Expressions and Assignments, Increment and Decrement Operator, Combined Operator.
- Selection and Switch, If statement Nested If, If else, Switch
- Repetition statement, While, Do While For Nested For statement.
- One dimensional array I.
- One dimensional array II.
- Two dimensional array I.
- Two dimensional array II.
- Structure I
- Structure II

Sources of Information

- Book
- Richard L. Halterman (2017), **Fundamentals of Programming C++.**

- Suggested references

  - Clifford A. Shaffer, Virginia Tech (2011), **Data Structures and Algorithm Analysis.**
  - James F. Korsh,Leonard J.Garrett (2008), **Data Structures, Algorithms, and Program style using(c)**,
  - Textbook. Michael T. Goodrich, Roberto Tamassia. David M. Mount (2007), **Data Structures and Algorithms in C++ .**

Assessment C++ Programming Language

- **Assessment of Learning:** The student will be evaluated based on homework, exams, and class participation.
- **Examination Information:** The exams will cover material that has been discussed in class or covered in the book.
- **Assignments:** Homework must be submitted at the beginning of class on the date which it is due. Late homework will be accepted only when special circumstances are approved by the instructor.
- **Terms examination: 40%**
  - Term Exams
  - Assignments
- **Final Exam: 60%**

Lecture one
Introduction C++ Programming Language

Computer and Development Tools

- A **computer program** is a sequence of instructions that dictate the flow of electrical impulses within a computer system. These impulses affect the computer's memory and interact with the display screen, keyboard, mouse, and perhaps even other computers across a network in such a way as to permit humans to perform useful tasks, solve high-level problems, and play games.

Computer and Development Tools

- Software must be stored in the computer's memory. These patterns of electronic symbols are best represented as a sequence of zeroes and ones, digits from the binary (base 2) number system. An example of a binary program sequence is: 10001011011000010001000001001110
- Software can be represented by printed words and symbols that are easier for humans to manage than binary sequences.
- Tools exist that automatically convert a higher-level description of what is to be done into the required lower-level code.

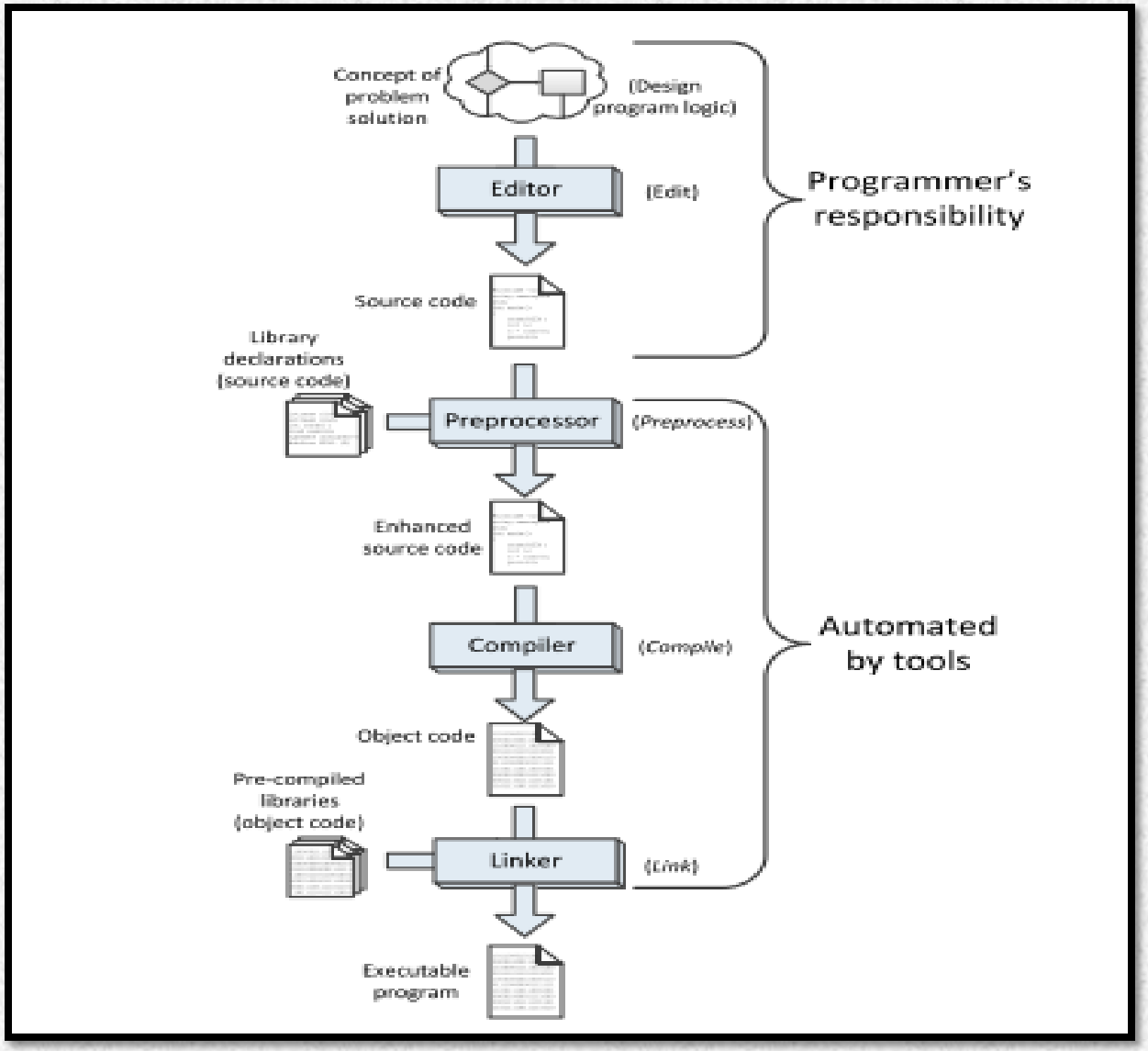
Computer and Development Tools

- Higher-level programming languages like C++ allow programmers to express solutions to programming problems in terms that are much closer to a natural language like English.
- Some examples of the more popular of the hundreds of higher-level programming languages that have been devised over the past 60 years include FORTRAN, COBOL, Lisp, Haskell, C, Perl, Python, Java, and C#.
- A C++ compiler is used to translate the C++ code into machine code. The higher-level language code is called source code. The compiled machine language code is called the target code. The compiler translates the source code into the target machine language.

Computer and Development Tools

- An editor allows the user to enter the program source code and save it to files. Most programming editors increase programmer productivity by using colors to highlight language features.
- The syntax of a language refers to the way pieces of the language are arranged to make well-formed sentences. programmers must follow strict syntax rules to create well-formed computer programs. Only well-formed programs are acceptable and can be compiled and executed. Some syntax-aware editors can use colors or other special annotations to alert programmers of syntax errors before the program is compiled.

Computer and Development Tools

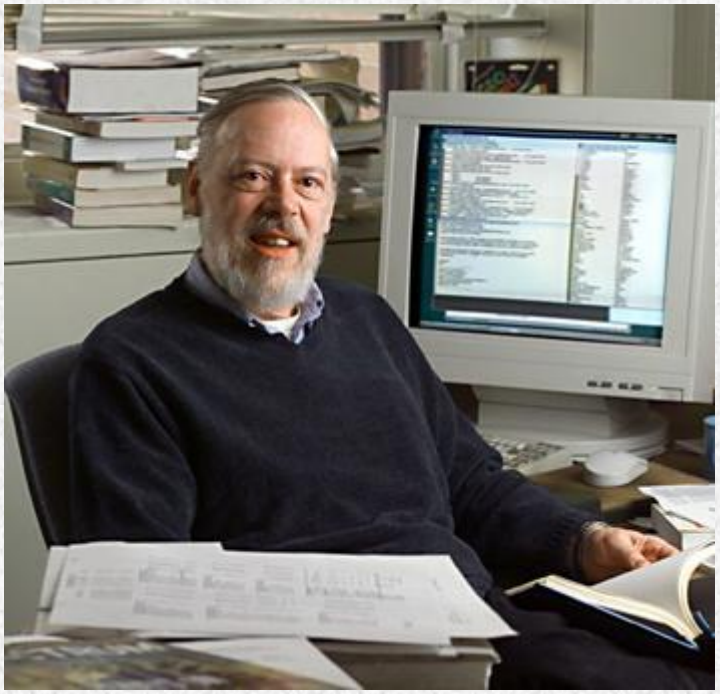


The Task of Programming

- **Machine language:** language that computers can understand; it consists of 1s and 0s
- **Interpreter:** program that translates programming language instructions one line at a time
- **Compiler:** translates entire program at one time
- **Run** a program by issuing a command to execute the program statements
- **Test** a program by using sample data to determine whether the program results are correct

What is C?

- Programming languages provide the appropriate formulas to define and use primitive data types.
- C is a rather old programming language (1972, Richie, Bell Labs)
- Originally designed as a systems software platform (OS and the like)
- Procedural, block oriented language (no object-oriented programming)



Dennis Ritchie
Inventor of the
C Programming Language

Why learn C++?

- Small, extensible language. Progenitor of many languages.
- It is everywhere! (portable)
- Many applications and much support due to its age and general use
- Many tools written to support C++ development.
- Close to the hardware, programmer manages memory
- Easy to interface with system devices/assembly routines
- Common embedded systems language
- Can be fast, efficient (most cited reason)

Why Learn C++?

- The following program is written in the C programming language:

```
#include <iostream.h>
main()
{
/* My first program */
cout<<“Hello World! \n“;
}
```

C++ Program

- C++ is case sensitive. All commands in C must be lowercase.
- C++ has a free-form line structure.
- End of each statement must be marked with a semicolon. Multiple statements can be on the same line.
- White space is ignored. Statements can continue over many lines.

C++ Program

The C program starting point is identified by the word main().

- This informs the computer as to where the program actually starts. The parentheses that follow the keyword main indicate that there are no arguments supplied to this program (this will be examined later on).
- The two braces, { and }, signify the begin and end segments of the program. In general, braces are used throughout C to enclose a block of statements to be treated as a unit. COMMON ERROR: unbalanced number of open and close curly brackets!

C++ Program

- The purpose of the statement #include <iostream.h> is to allow the use of the cout statement to provide program output. For each function built into the language, an associated header file must be included.
- cout is actually a function (procedure) in C++ that is used for printing variables and text. This has to do with the \ and % characters. These characters are modifiers, and for the present the
- \ followed by the n character represents a newline character.

C++ Program

- Thus the program prints

  Hello World!

  And the cursor is set to the beginning of the next line.

- Comments can be inserted into C++ programs by bracketing text with the /* and */ delimiters. Comments are useful for a variety of reasons. Primarily they serve as internal documentation for program structure and functionality.

Why Use Comments?

- Documentation of variables and functions and their usage
- Explaining difficult sections of code
- Describes the program, author, date, modification changes, revisions…

** Best programmers comment as they write the code, not after the fact.

Header Files

- Header files contain definitions of functions and variables which can be incorporated into any C++ program by using the pre-processor #include statement.
- Standard header files are provided with each compiler, and cover a range of areas: string handling, mathematics, data conversion, printing and reading of variables, etc.
- For example, to use the function cout in a program, the line #include <iostream.h> • should be at the beginning of the source file, because the declaration for cout is found in the file iostream.h.

Header Files

- All header files have the extension .h and generally reside in the /usr/include subdirectory.

#include <string.h>

#include <math.h>

#include "mylib.h“

- The use of angle brackets <> informs the compiler to search the compiler’s include directories for the specified file. The use of the double quotes "" around the filename informs the compiler to start the search in the current directory for the specified file.

Header Files

- All header files have the extension .h and generally reside in the /usr/include subdirectory.

```
#include <string.h>
#include <math.h>
#include "mylib.h“
```

- The use of angle brackets <> informs the compiler to search the compiler’s include directories for the specified file. The use of the double quotes "" around the filename informs the compiler to start the search in the current directory for the specified file.