Fundamental of Programming (C)

Lecturer: Omid Jafarinezhad

Lecture 1

Introduction and Brief History

Outline

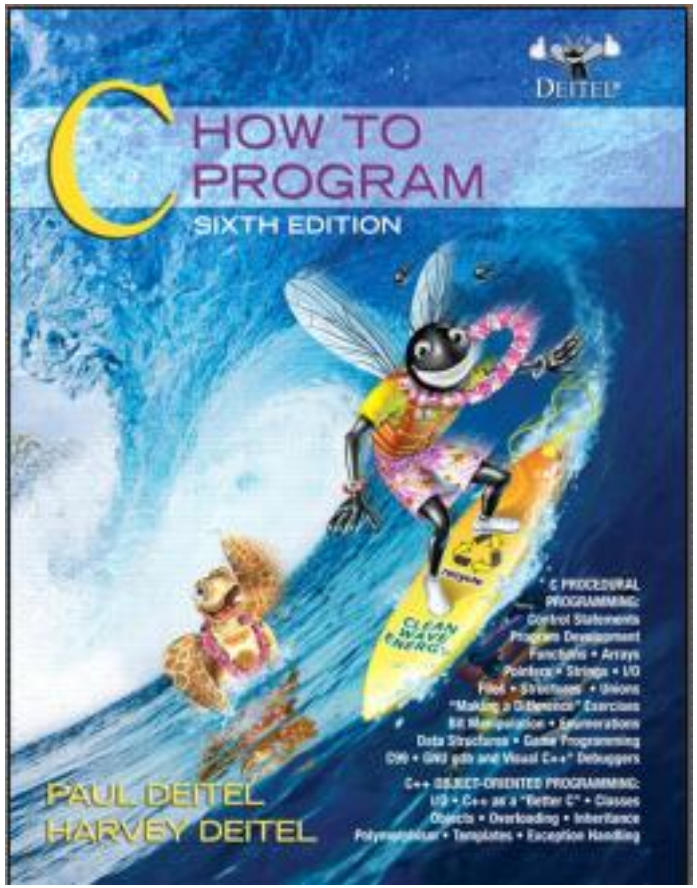
- Review Of Course Materials
- Grading Policy
- An Overview Of Computer
  - Computer Components
    - Hardware
    - Software
- Introduction To Programming
  - programming paradigm
  - Machine Languages
  - Assembly Languages
  - High-Level Languages
- History Of C/C++
- Typical C Program Development Environment
  - Compilation Process

Review Of Course Materials

- Computer number format
- Data Types, Variables, Operators, Input/output
- Algorithm and Pseudo code
- Functions
- Strings and Pointers
- Arrays, Structures
- Files
- Object-Oriented Programming

Core reference

- Deitel, C How to Program, Sixth edition, Pearson Education



Grading policy

- Quizzes: 10%
- Assignments: 20%
- Midterm exam: 20%
- Final exam: 35%
- Final programming projects: 15%
- Many bonus chances …

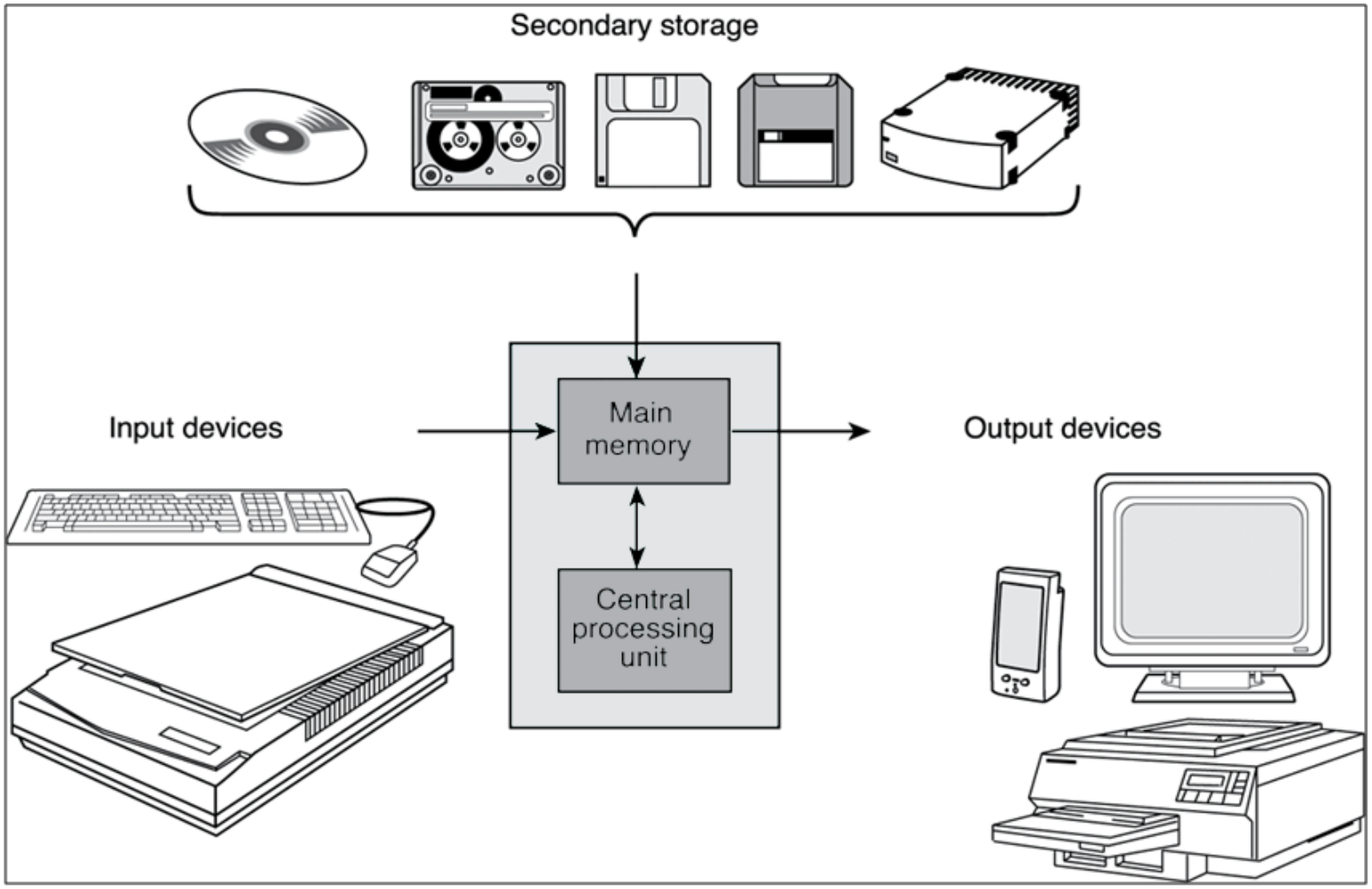
An Overview Of Computer

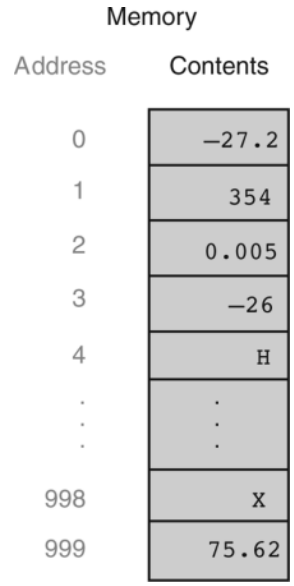
- Computer:
  - programmable general purpose machine
    - can not do anything without a Program
  - receives input
    - letters, numbers, images
  - processes and stores input
  - Provides output in a useful format

Computer Components

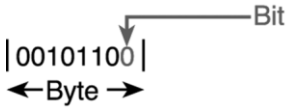
- Hardware
  - the physical parts or components of computer such as monitor, keyboard, hard disk, mouse, etc.
- Software
  - instructions you write to command computers to perform actions on hardware

Overview of Computer Hardware





Ordered sequence of storage location (memory cell)



logical units

- Regardless of differences in physical appearance, virtually every computer may be envisioned as divided into six logical units or sections

- Input unit : receiving section obtains information (data and computer programs) from input devices and places it at the disposal of the other units so that it can be processed

  – input devices: keyboards and mouse

Output unit

- This shipping section takes information that the computer has processed and places it on various output devices to make it available for use outside the computer

Memory unit

- rapid-access, relatively low-capacity
- warehouse section retains information that has been entered through the input unit, making it immediately available for processing when needed
- The memory unit also retains processed information until it can be placed on output devices by the output unit
- Information in the memory unit is volatile—it's typically lost when the computer's power is turned off
- The memory unit is often called either memory or primary memory

Arithmetic and logic unit (ALU)

- manufacturing section performs calculations
  - addition, subtraction, multiplication and division
  - It also contains the decision mechanisms that allow the computer, for example, to compare two items from the memory unit to determine whether they're equal

- the ALU is usually implemented as part of the next logical unit, the CPU

Central processing unit (CPU)

- administrative section coordinates and supervises the operation of the other sections
  - tells the input unit when to read information into the memory unit
  - tells the ALU when information from the memory unit should be used in calculations
  - tells the output unit when to send information from the memory unit to certain output devices

- Multiprocessors computers have multiple CPUs and, hence, can perform many operations simultaneously
  - A multi-core processor implements multiprocessing on a single integrated circuit chip
    - a dual-core processor has two CPUs
    - a quad-core processor has four CPUs

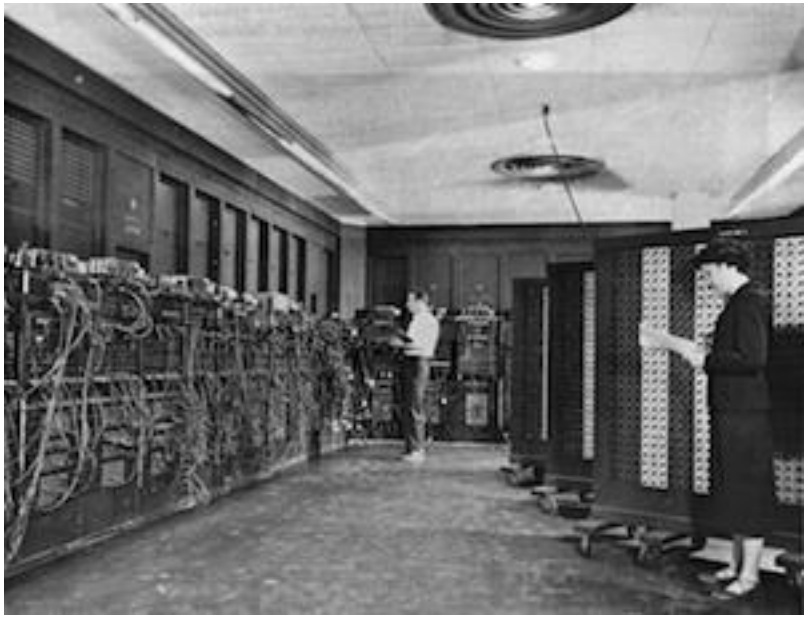
Secondary storage unit

- long-term
- high-capacity warehousing section
- Programs or data not actively being used by the other units normally are placed on secondary storage devices until they're again needed
- Information on secondary storage devices is said to be persistent—it is preserved even when the computer's power is turned off
  - Hard drives, CDs, DVDs and flash drives

History - First Generation Computers

- Mid-1940s
- used vacuum tubes
- huge and complex

The ENIAC, weighing 30 tons, using 200 kilowatts of electric power and consisting of 18,000 vacuum tubes



History - Second Generation Computers

- 1955 – 1960
- The invention of transistor
- The era of miniaturization begins.

History - Third Generation Computers

- 1960s
- the Integrated Circuits , also known as microchips
- silicon chips containing multiple transistors

History - Fourth Generation Computers

- 1971 – present
- large-scale integration or LSI
  - 1000 devices per chip)
- very large-scale integration or VLSI
  - 10000 devices per chip)
- …

Overview of Computer Software

- Operating System (OS)
  - the collection of computer programs that control the interaction of the user and the computer hardware.
  - E.g. Windows, Unix
- Application Software
  - Programs developed to assist a computer user in accomplishing specific tasks.
  - E.g. Microsoft Word
- In order to create new application software, we need to write lists of instruction (program) to the computer to execute

Programming Language

- The defining feature of modern computers which distinguishes them from all other machines is that they can be programmed

- Programming is instructing a computer to do something for you with the help of a Programming Language

- A programming language contains instructions for the computer to perform a specific action or a specific task:
  - Display “I like programming”
  - Display the current time

Programming Language

- Programming Language is a Formal Language used to communicate to a computer
  - Very specific (one word means one thing – context free) since to 'talk' to a computer; to instruct a computer; our commands must be 100% clear and correct

- The description of a programming language is usually split into the two components of syntax (form) and semantics (meaning)

- A programming paradigm is a fundamental style of computer programming :
  - Functional : tell what to do but not how (sum [1...10])
  - Imperative : describing step by step
  - Object-Oriented and Logical Programming

Programming Language

- Special-purpose : is design for a particular type of application
  - Structured Query Language (SQL)

- General-purpose : can be used to obtain solutions for many types of problems.
  - Machine Languages
  - Assembly Languages
  - High-Level Languages

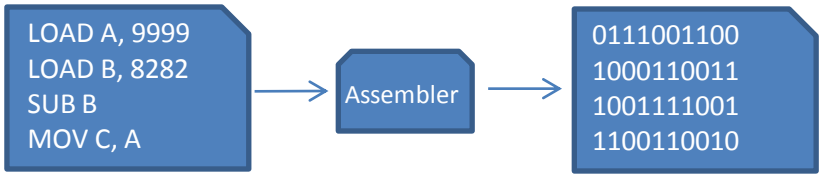
Machine Language

- The only language that the processor actually understands
- Consists of binary codes: 0 and 1
  - Example: 00010101
    11010001
    01001100
- Each of the lines above corresponds to a specific task to be done by the processor
- Programming in machine code is difficult and slow since it is difficult to memorize all the instructions
- Mistakes can happen very easily
- Processor and Architecture dependent (different machine language for different type of CPU) – not portable

Assembly Language

- Enables machine code to be represented in words and numbers
- Example of a program in assembly language:

```
LOAD A, 9999
LOAD B, 8282
SUB B
MOV C, A
```



- Easier to understand and memorize (called Mnemonics), compared to machine code but still quite difficult to use
- Cannot be processed directly by a computer, must be converted to machine language using assemblers
- Processor and Architecture dependent – not portable

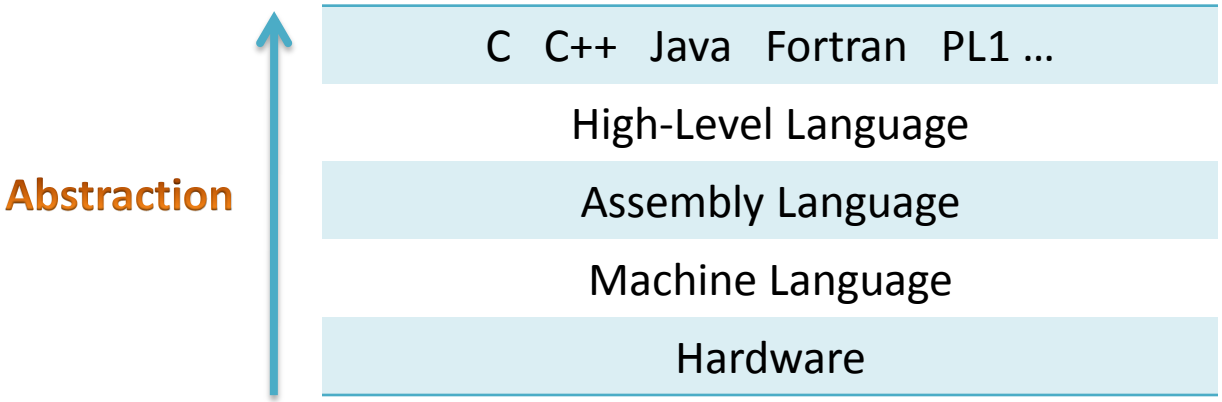
High-Level Language

- Machine independent programming language that combines algebraic expression and English words
- Example:

  c = b - a

- Processor independent - the same code can be run on different processors
- Examples: Basic, Fortran, Pascal, Cobol, C, C++, Java
- High level language needs to be translated (compiled) to machine code by a program called compiler so that it can be executed by the processor

Programming Language Abstraction



C History

- BCPL ,1967, Martin Richards
  - writing operating-systems software and compiler
- B, 1969, Ken Thomson
  - based on BCPL
- C, 1972, Dennis Ritchie
  - based on BCPL and B
  - at Bell Laboratories
  - originally implemented on a DEC PDP-11

C History

- In 1983, the American National Standards Institute (ANSI) established a committee to provide a modern, comprehensive definition of C. The resulting definition, the ANSI standard, or ANSI C, was completed late 1988
  - updated in 1999

- Because C is a hardware-independent, widely available language, applications written in C can run with little or no modifications on a wide range of different computer systems
  - Portable programs

C – An Imperative Language

- C is a highly imperative formal language
  - We must tell it exactly how to do what
  - the means and functions to use
  - which libraries to use
  - when to add a new line
  - when an instruction is finished
  - in short: everything and anything…
- *filename.c*

C++ Programming Language

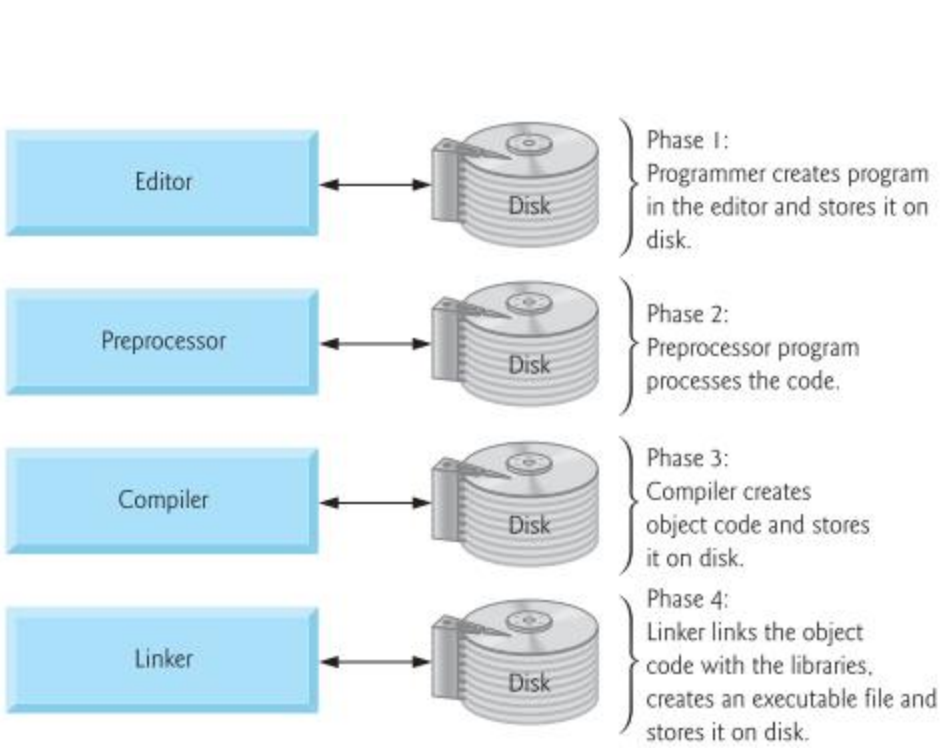
- early 1980s, Bjarne Stroustrup
  - at Bell Labroratory
  - C++ a superset of C
  - object-oriented programming
    - Objects are essentially reusable software components that model items in the real world

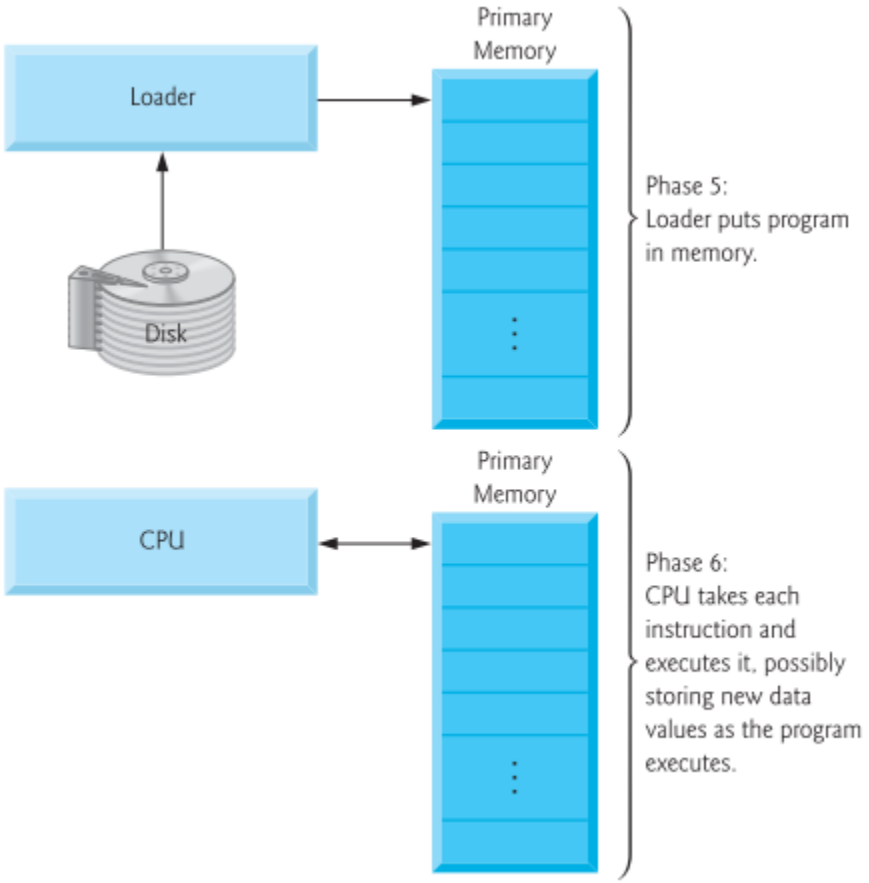
- *filename.cpp*

Typical C Program Development Environment

- C systems generally consist of several parts:
  - a program development environment
  - the language
  - the C Standard Library

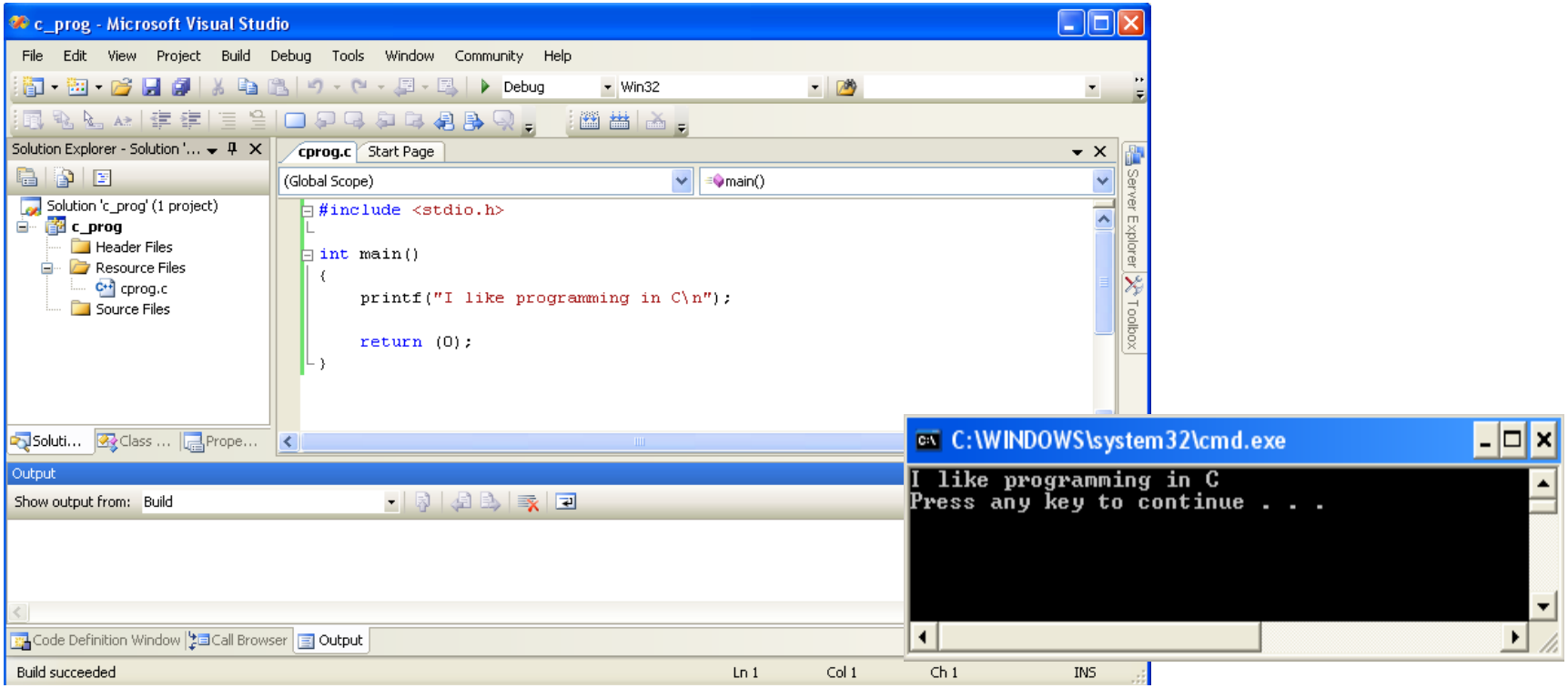
- C programs typically go through six phases to be executed:
  - edit, preprocess, compile, link, load and execute


Editor
Disk
Phase 1:
Programmer creates program in the editor and stores it on disk.
Preprocessor
Disk
Phase 2:
Preprocessor program processes the code.
Compiler
Disk
Phase 3:
Compiler creates object code and stores it on disk.
Linker
Disk
Phase 4:
Linker links the object code with the libraries, creates an executable file and stores it on disk.

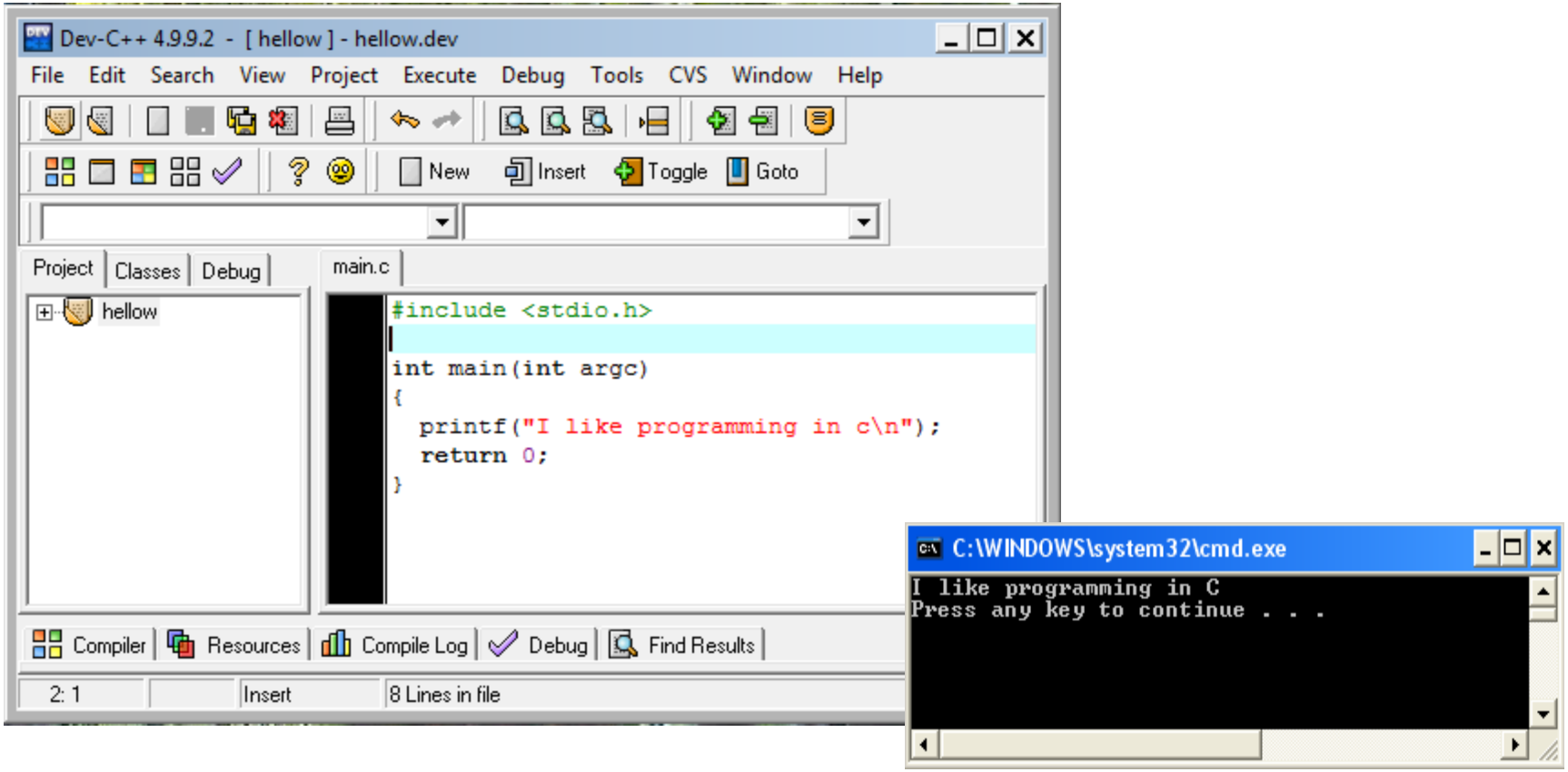

Loader
Primary Memory
Disk
Phase 5:
Loader puts program in memory.
CPU
Primary Memory
Phase 6:
CPU takes each instruction and executes it, possibly storing new data values as the program executes.

Microsoft Visual Studio

- Editing a file with an editor program
- Integrated Development Environment (IDE)



Dev-C++


Dev-C++ 4.9.9.2 - [hellow] - hellow.dev
File Edit Search View Project Execute Debug Tools CVS Window Help
New Insert Toggle Goto
Project Classes Debug
hellow
main.c
#include <stdio.h>
int main(int argc)
{
printf("I like programming in c\n");
return 0;
}
Compiler Resources Compile Log Debug Find Results
2: 1 Insert 8 Lines in file
C:\WINDOWS\system32\cmd.exe
I like programming in C
Press any key to continue . . .

Preprocessor And compiler

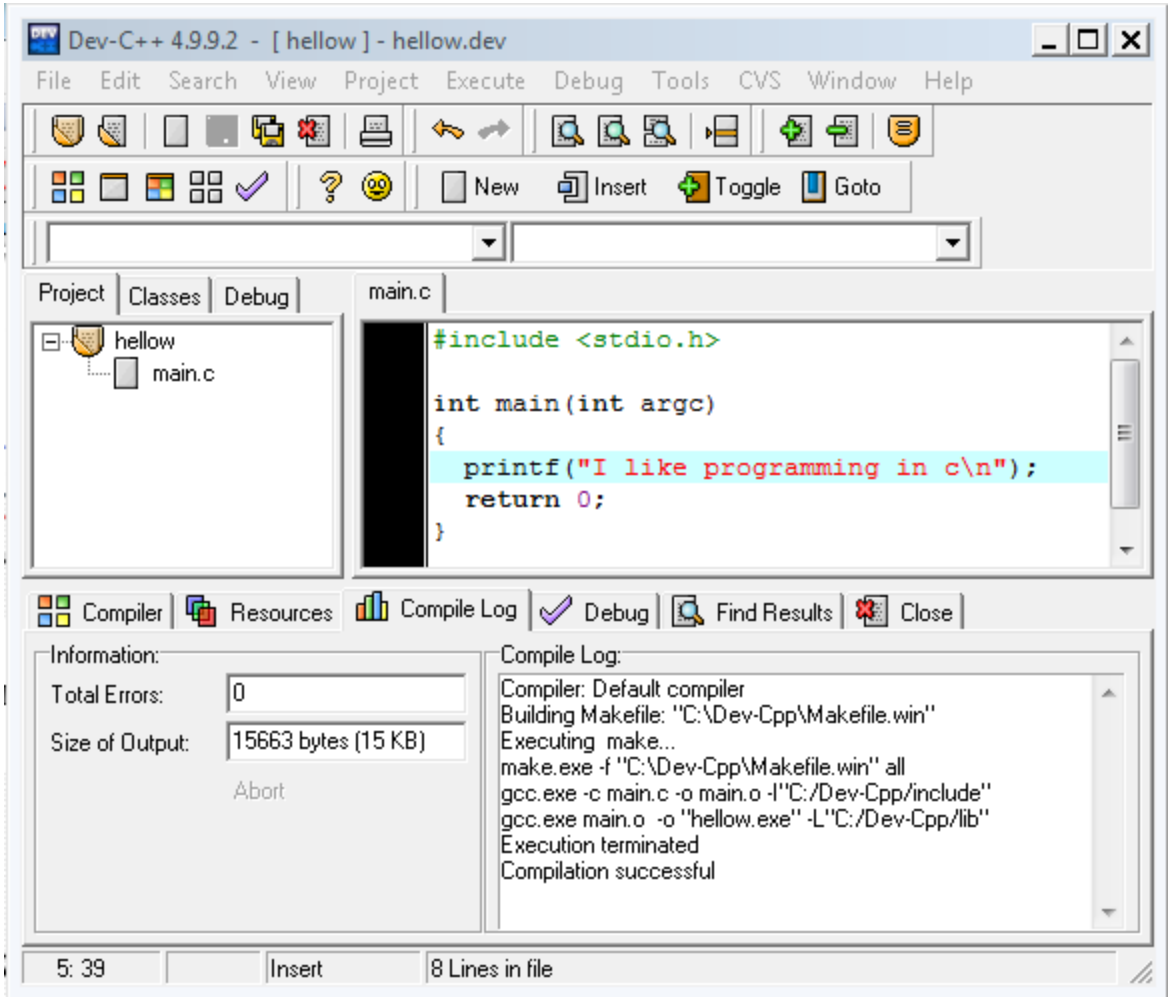
- a preprocessor program executes automatically before the compiler's translation phase begins
  - The C preprocessor obeys special commands called preprocessor directives, which indicate that certain manipulations are to be performed on the program before compilation

- The compiler translates the C program into machine language-code (object code)

Linking, Loading And Execution

- C programs typically contain references to functions defined elsewhere, such as in the standard libraries or in the private libraries
  - A linker links the object code with the code for the missing functions to produce an executable image
- Before a program can be executed, the program must first be placed in memory
  - This is done by the loader, which takes the executable image from disk and transfers it to memory
  - Additional components from shared libraries that support the program are also loaded
- Finally, the computer, under the control of its CPU, executes the program one instruction at a time

Compile log


Dev-C++ 4.9.9.2 - [hellow] - hellow.dev
File Edit Search View Project Execute Debug Tools CVS Window Help
New Insert Toggle Goto
Project Classes Debug
hellow
main.c
main.c
#include <stdio.h>
int main(int argc)
{
printf("I like programming in c\n");
return 0;
}
Compiler Resources Compile Log Debug Find Results Close
Information:
Total Errors: 0
Size of Output: 15663 bytes (15 KB)
Abort
Compile Log:
Compiler: Default compiler
Building Makefile: "C:\Dev-Cpp\Makefile.win"
Executing make...
make.exe -f "C:\Dev-Cpp\Makefile.win" all
gcc.exe -c main.c -o main.o -I"C:/Dev-Cpp/include"
gcc.exe main.o -o "hellow.exe" -L"C:/Dev-Cpp/lib"
Execution terminated
Compilation successful
5: 39
Insert
8 Lines in file

Common Problems of Programming

- Usability
  - Your program is too complicated or too simple to be useful to most people

- Maintainability
  - Other people, or yourself at a later time can't easily understand the programming behind your program. This means your project won't grow and become all it's capable of being

Summary

- Computer Components
  - Hardware
    - Logical Computer Organization: Input unit, Output unit, Memory unit, ALU, CPU, Secondary storage unit
    - Generations Of Computer Hardware: vacuum tube, transistor, IC, LSI , V LSI
  - Software
    - Operating System
    - Application Software
- Programming Languages
  - programming paradigm: Functional, Imperative, Object-Oriented, Logical
  - Machine Languages: language of processor; represented by 0 and 1
  - Assembly Languages: represented in words and numbers
  - High-Level Languages: machine independent
- History Of C/C++ : based on BCPL, B; imperative language
- Typical C Program Development Environment
  - Compilation Process : edit, preprocess, compile, link, load and execute