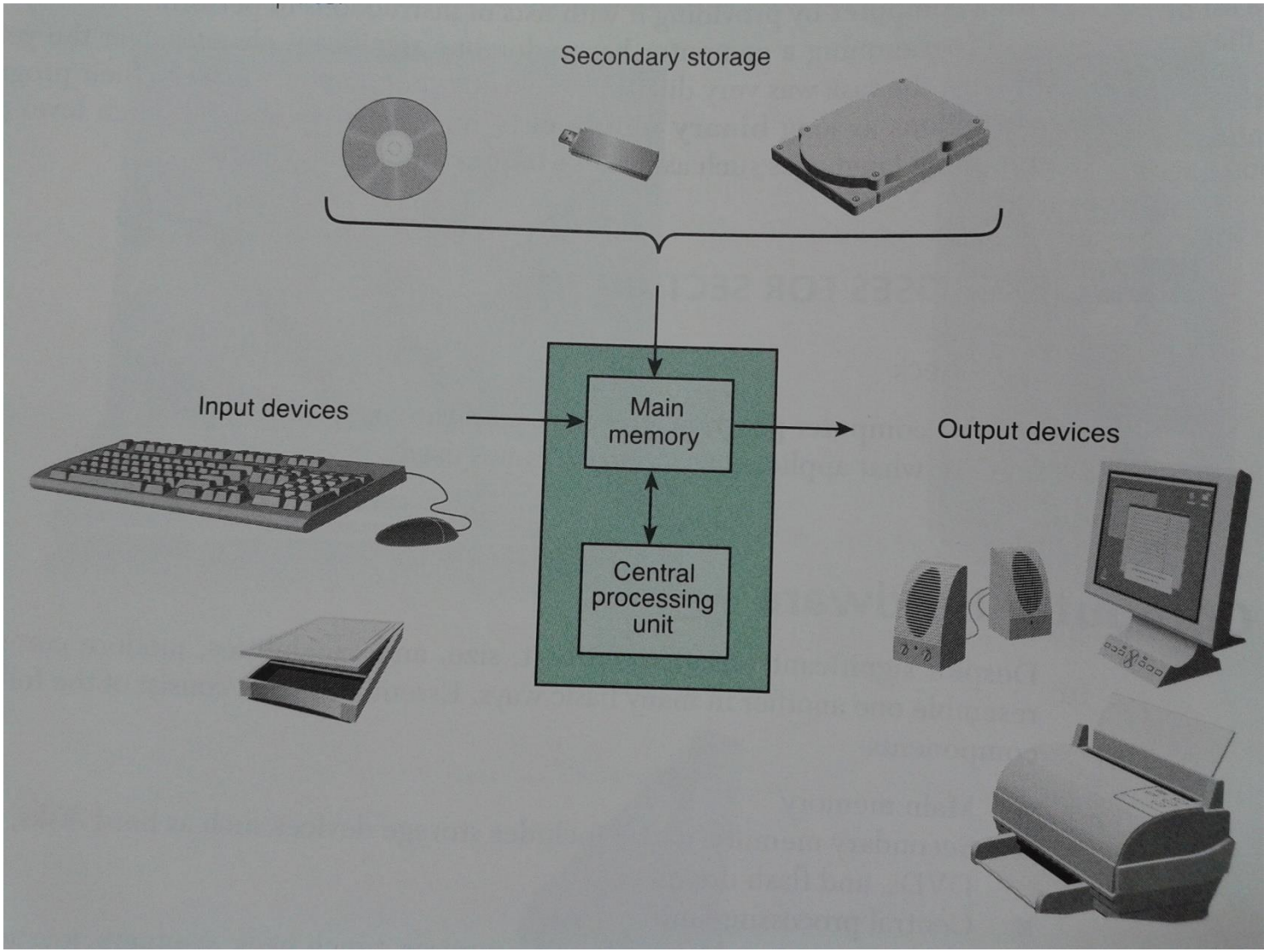
Chapter 1: Introduction

- In this chapter you will learn about:
  - Overview of Computer Component
  - Overview of Programming
  - Programming Language
  - C Programming Language

Overview of Computer Component

- Computer Hardware
- Computer Software

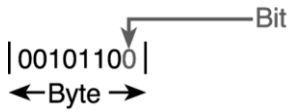
Overview of Computer Hardware



Memory

Address	Contents
0	–27.2
1	354
2	0.005
3	–26
4	H
. . .	. . .
998	X
999	75.62

Ordered sequence of storage location (memory cell)



Overview of Computer Software

- Operating System (OS)
  - the collection of computer programs that control the interaction of the user and the computer hardware.
- 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

- ***Programming*** is instructing a computer to do something for you with the help of a Programming Language
- A program contains ***instructions*** for the computer to perform a specific action or a specific task:
  - Display the current time
  - Calculate the salary

Programming

- The two roles of programming:
  - **Technical**: It instructs the computer to perform tasks.
  - **Conceptual**: It is a framework within which we organize our ideas about things and processes.
- In programming, we deal with two kind of things:
  - **Data** - representing '*objects*' we want to *manipulate*
  - **Procedures** -'*descriptions*' or '*rules*' that define *how* to manipulate data

Programming Language

- Programming language, like natural language, consists of syntax and semantics.
  - Syntax - Syntax are the rules to join words together in forming a correct ***expression*** or phrase.
  - Semantics – the meaning of the language within a given context

Programming Language

- Programming language is different from natural language (our daily spoken language) in terms of:
  - Natural language does not have to be 100% correct but still understandable.
  - In natural language, one word can have different meaning depending on the situation.
    - For example, the word *PUNCH.*

    In a restaurant

    "I'd like to order a glass of fruit punch, please."

    In an investigation, a police may ask

    "Who punched you?"
- Programming language is a formal language because it is very specific. One word has only one meaning regardless of the contexts. i.e. context free.

Programming Language

- Programming language can be classified into special-purpose and general-purpose programming languages.
- Special-purpose is designed for a particular type of application
  - Structured Query Language (SQL)
- General-purpose can be used to obtain solutions for many types of problems.
  - Visual Basic

Programming Language

- Programming language can be seen as consisting of the following three types.
  - Machine Languages
  - Assembly Languages
  - High-Level Languages

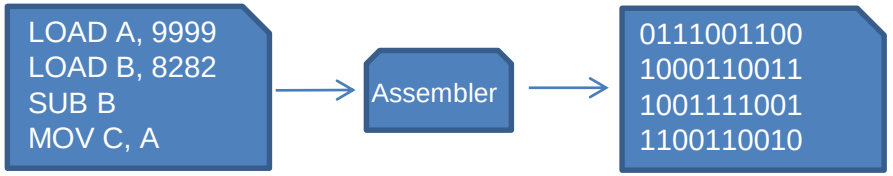
Machine Language

- The only language that the computer 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.
- Machine language is processor and architecture dependent and not portable i.e. different machine language is used for different type of processor.

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, 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
- It is also processor and architecture dependent.

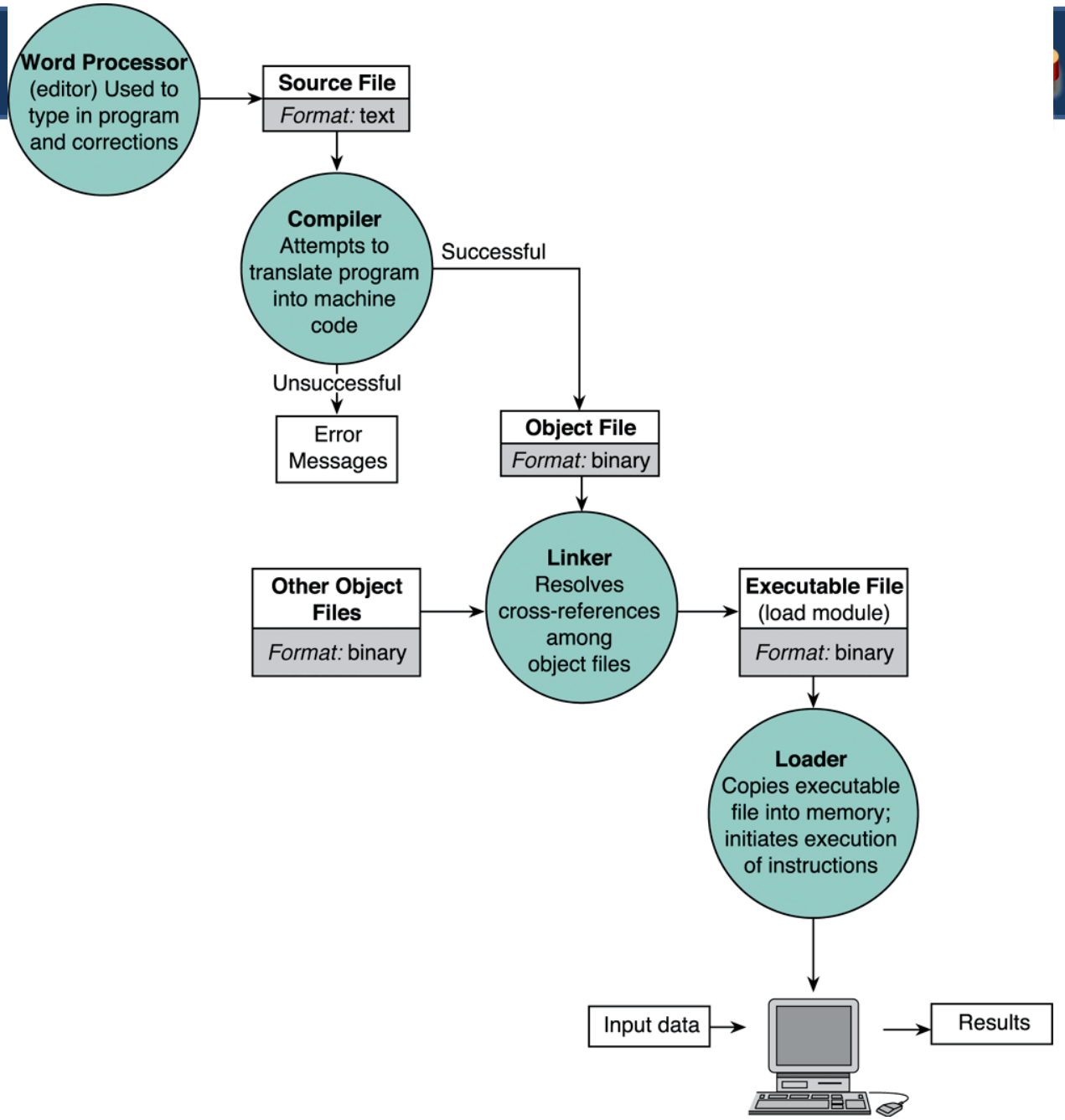
High-Level Language

- Combines algebraic expression and English words.
  - Example:

$$c = b - a$$

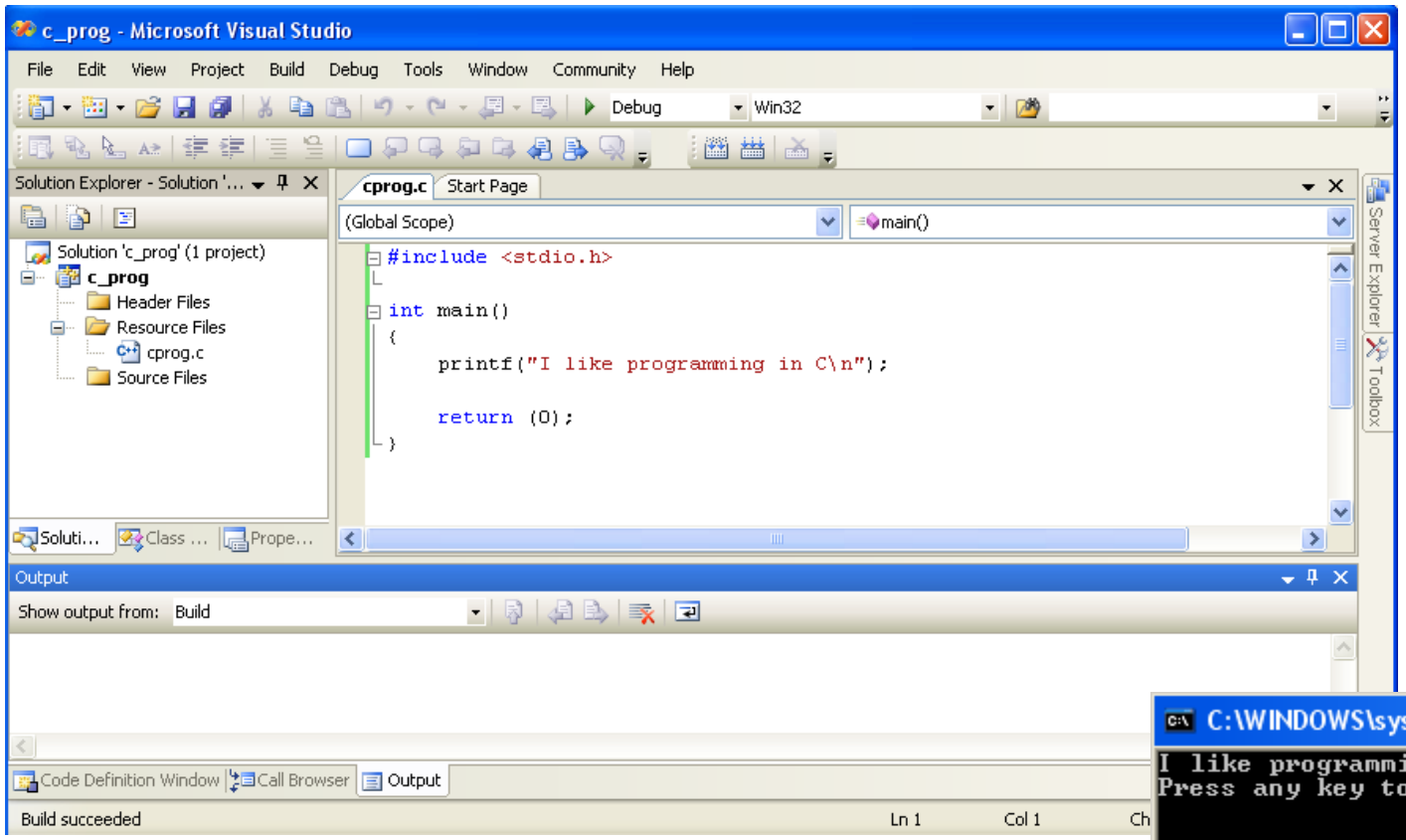
- Examples of high level programming languages: ***Basic, Fortran, Pascal, Cobol, C, C++, Java***
- High level language needs to be translated into machine code by a program called **compiler** so that it can be **executed** by the processor.
- High level language is processor **independent** i.e. the same code can be run on different processors.

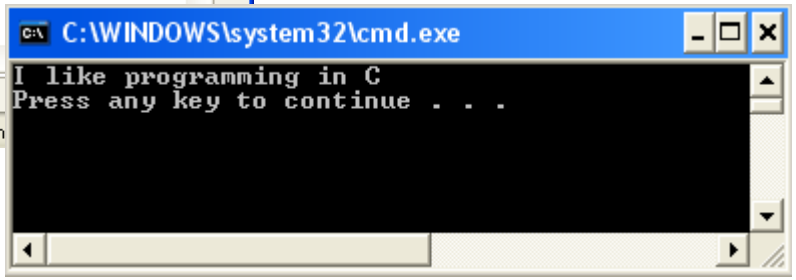
Writing, translating, and executing a program



Integrated Development Environment (IDE)

- A package that combines a simple word processor with a compiler, linker and loader.





C Programming Language

- Why 'C' ? Because based on '**B**'; developed at **B**ell Laboratories
- Developed by **Dennis Ritchie** at Bell Laboratories in the 1960s
- In cooperation with **Ken Thomson** it was used for Unix systems
- Initially, the C Language was only vaguely defined, not standardized, so that almost everyone had his own perception of it, to such an extent that an urgent need for a **standard code** was creeping up

C Programming Language cont…

- In 1983, the American National Standards Institute (ANSI) set up X3J11, a Technical Committee to draft a proposal for the ANSI standard, which was approved in 1989 and referred to as the ANSI/ISO 9899 : 1990 or simply the **ANSI C**, which is now the global standard for **C**.

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…
- Program written using C Language is stored with the .c extension, for example
  - *filename.c*

C++ Programming Language

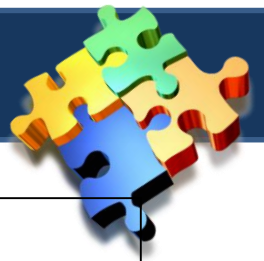
- Created by Bjarne Stroustrup at Bell Labroratories in early 1980s
- Is an extension of the C language
- C++ contains all features of C and new features (C++ a superset of C)
- C++ makes object-oriented programming possible.
- Program written using C++ Language is stored with the .cpp extension, for example
  - *filename.cpp*

A Simple Program in C

```
#include <stdio.h>

int main(void)
{
      printf("I like programming in C.\n");
      return (0);
}
```

A Simple Program in C - explanation



```
#include <stdio.h>

int main(void)
{
        printf("I like programming in C.\n");

        return (0);
}
```

stand**d**ard Library, **i**nput-**o**utput, **h**eader-file

Beginning of the program

Start of Segment

Function for printing text

Insert a new line

End of statement

End of Segment

C Output

I like programming in C.

Summary

- In this chapter, we have looked at the following
  - Components of a computer
  - The meaning of programming and programming language
  - Types of programming languages
  - Origin of C programming language
  - the C language syntax to print a line of text onto the screen of our computer.