上海交通大學
SHANGHAI JIAO TONG UNIVERSITY

ParisTech
INSTITUT DES SCIENCES ET TECHNOLOGIES
PARIS INSTITUTE OF TECHNOLOGY

CS139
C Programming and Algorithm Analysis

Instructor: Jialiang Lu

Email: jialiang.lu@sjtu.edu.cn

Office: SPEIT Building 429

Course Information:

- Course Page:
  - http://moodle.speit.sjtu.edu.cn/ (C Programming)
  - Assignments (HW): http://wirelesslab.sjtu.edu.cn:8088/jol/
- Assistant:
  - Mingcheng HE ():
  - mingcheng.he@sjtu.edu.cn
- QQ group:

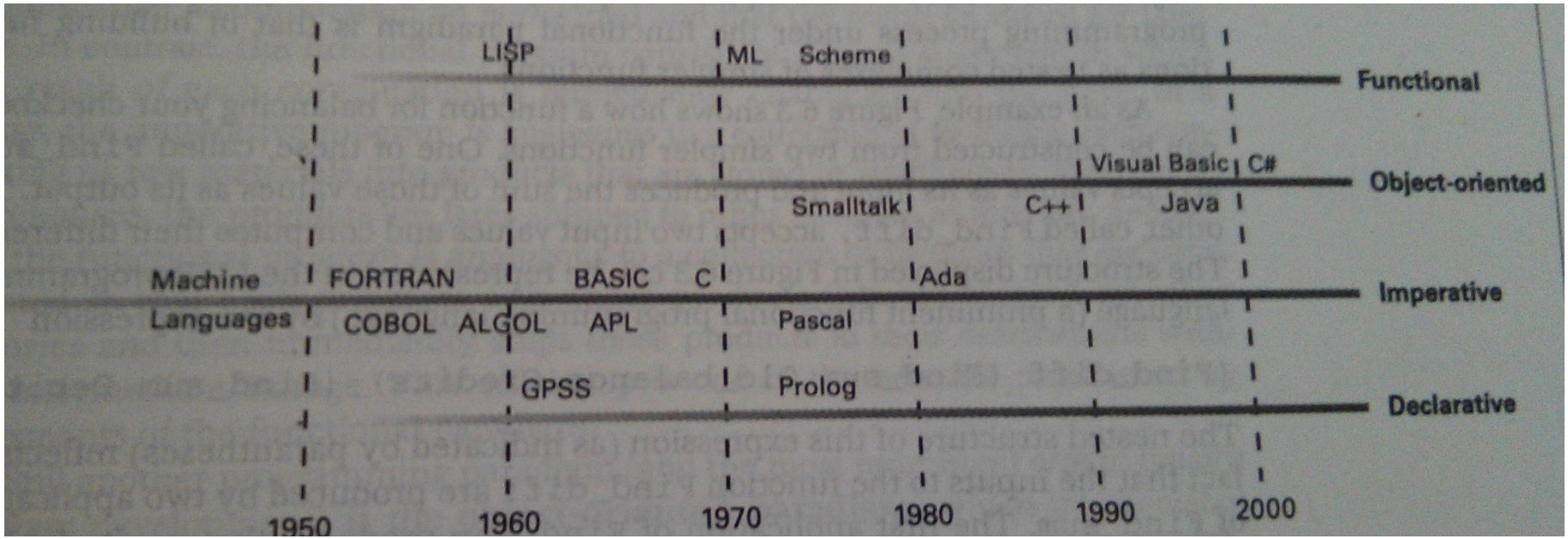
Programming in C

- Reference books:
  - *The C Programming Language* by Brian W. Kernighan and Dennis M. Ritchie
  - *Programming in C (3rd Edition)* by Stephen G. Kochan.
- www.learn-c.org online training
- Code::Blocks: IDE for C, C++ and Fortain , 16.01
- gcc: http://gcc.gnu.org/

ParisTech
INSTITUT DES SCIENCES ET TECHNOLOGIES
PARIS INSTITUTE OF TECHNOLOGY

WHAT YOU KNOW ABOUT PROGRAMMING?

Programming Paradigms



Programming Languages

- Funders: FORTRAN (1954), LISP (1958) and COBOL (1959)
- New arrivals: Simula (1962), Forth (1968), SQL(1970), SmallTalk (1972), Prolog (1972) et TEX (1977)
- Descendents: C (1972), C++ (1983), Postscript (1983), VHDL (1987), Python (1991), Java (1995), C# (2002), Go (2009),etc.

Programming Paradigms

- Imperative paradigm
  - A sequence of commands that, when followed, manipulate data to produce the desired result
- Declarative paradigm
  - Describe the problem rather than an algorithm
- Functional paradigm
  - A program is viewed as en entity that accepts inputs and produces outputs
- Object-Oriented paradigm/programming (OOP)

Statements

- A program consists of a collection of statements
  - Declarative statements
    - `int  Height, Width;`
  - Imperative statements
    - `Z = X + Y;`
    - `if .. else..`
  - Comments
    - //This is an addition
    - /* Declaration of variable*/

Variable and Data Types

- **Variable** is used by high-level programming language to be referenced in main memory.
  - Identified by declarative statements
    - `int  Height, Width;`
- **Data type**
  - the manner in which the data item is encoded
  - The operations that can be performed on the data.

Abstract Data Types

Abstract Data Type (ADT): a definition for a data type solely in terms of a set of values and a set of operations on that data type.

Each ADT operation is defined by its inputs and outputs.

Encapsulation: Hide implementation details.

Data Structure

- A data structure is the physical implementation of an ADT.
  - Each operation associated with the ADT is implemented by one or more subroutines in the implementation.
- Data structure usually refers to an organization for data in main memory.
- File structure is an organization for data on peripheral storage, such as a disk drive.

Logical vs. Physical Form

Data items have both a logical and a physical form.

Logical form: definition of the data item within an ADT.

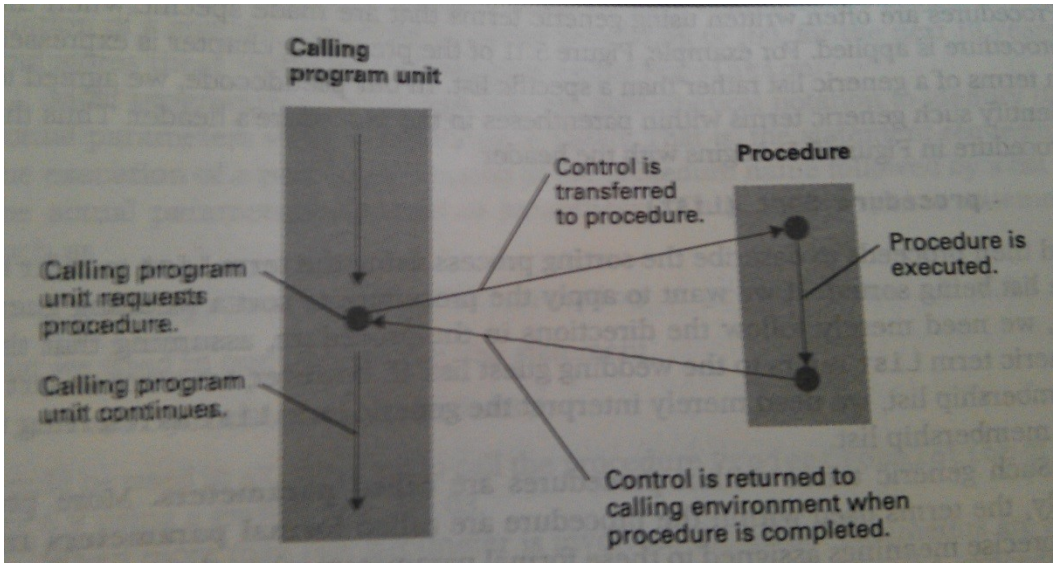
- Ex: Integers in mathematical sense: +, -

Physical form: implementation of the data item within a data structure.

- Ex: 16/32 bit integers, overflow.

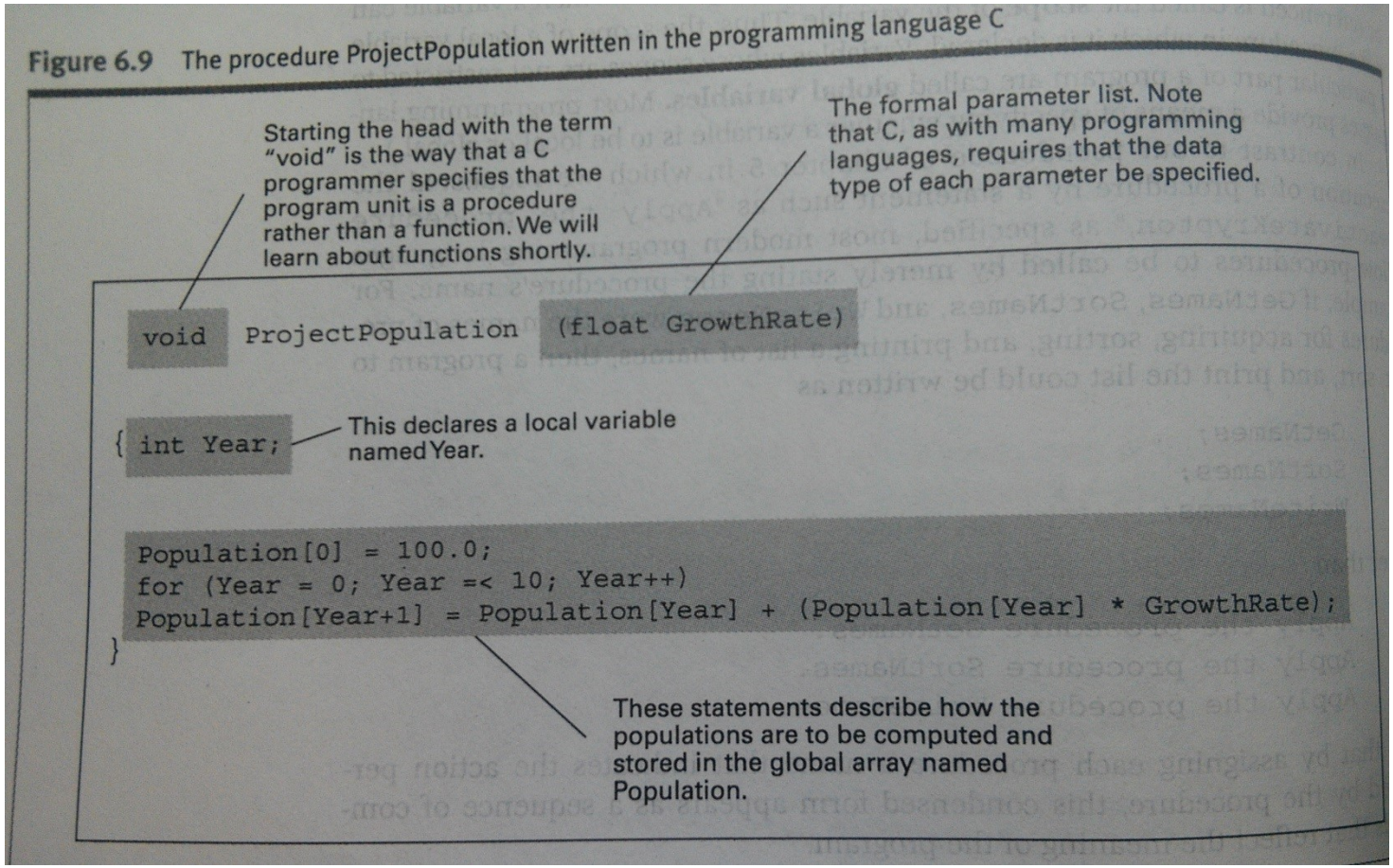
Procedure Units

- A **procedure** is a set of instructions for performing a task that can be used as an abstract tool by other program units.



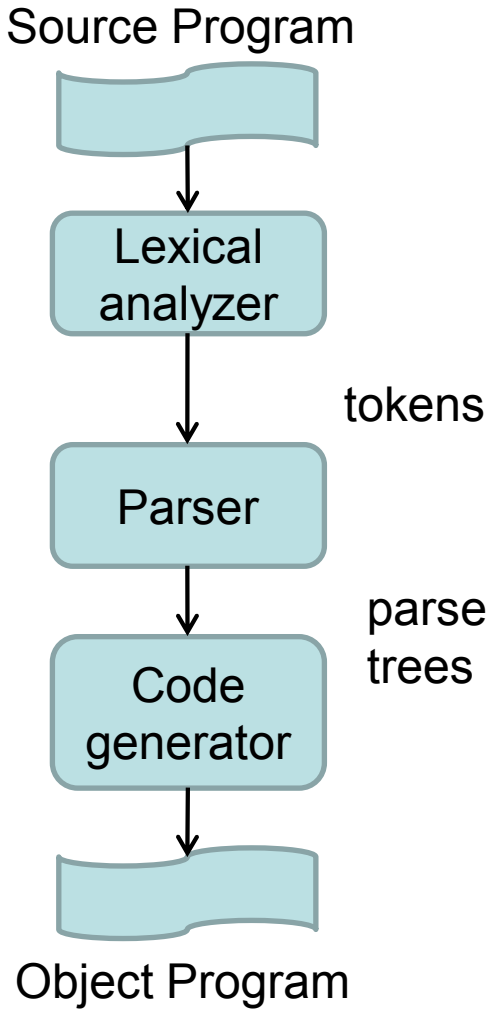
An example of Procedure

Figure 6.9 The procedure ProjectPopulation written in the programming language C

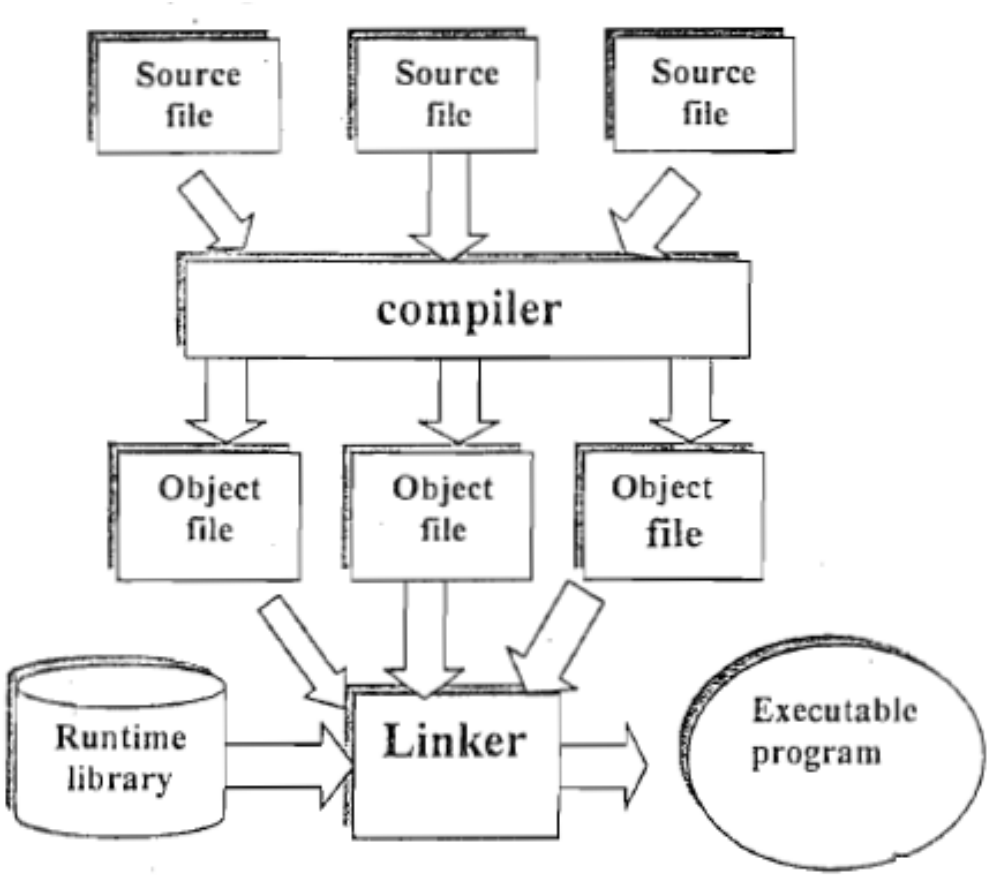


The translation process

- From source program to object program
  - .c → .o
  - .java → .class
- Lexical analyzer
  - find token
- Parser
  - Group units into statements
  - Generating a parse tree
- Code generator
  - Code optimization



Compiler and Linker



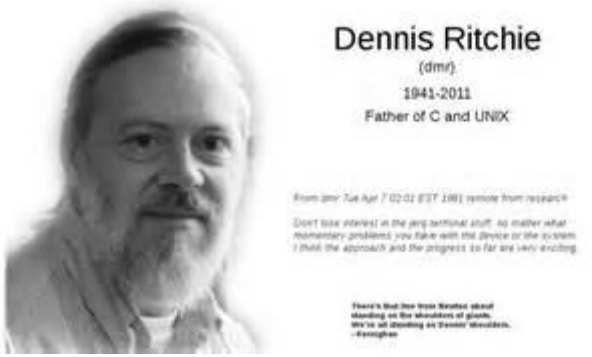
Why you have to learn C

- Steve Yegge (famous for his blog): Because for all practical purposes, every computer in the world you'll ever use is a von Neumann machine, and C is a lightweight, expressive syntax for the von Neumann machine's capabilities.

SO YOU THINK YOU KNOW C?

A little history:

- 1970's
  - Unix
  - C, from BCPL (Thompson and Ritchie)
- C programming Language
  - Widely used like the others: Fortran, Pascal
  - Main form of language for system programming
  - Available on any machine with C compiler and library



Some Characteristics:

- Scope:
  - Intended:
    - HPC, OS, general programming
    - Typically, long developing cycle
    - very platform dependent
  - Not intended: web, scripting, data processing
- Features:
  - Close interaction with system
    - Standard library (user-level only),
    - no other fancy stuff: networking, memory, graphics
  - Extensive use of pointers
  - Procedure programming, strictly pass by value

Sample 1:

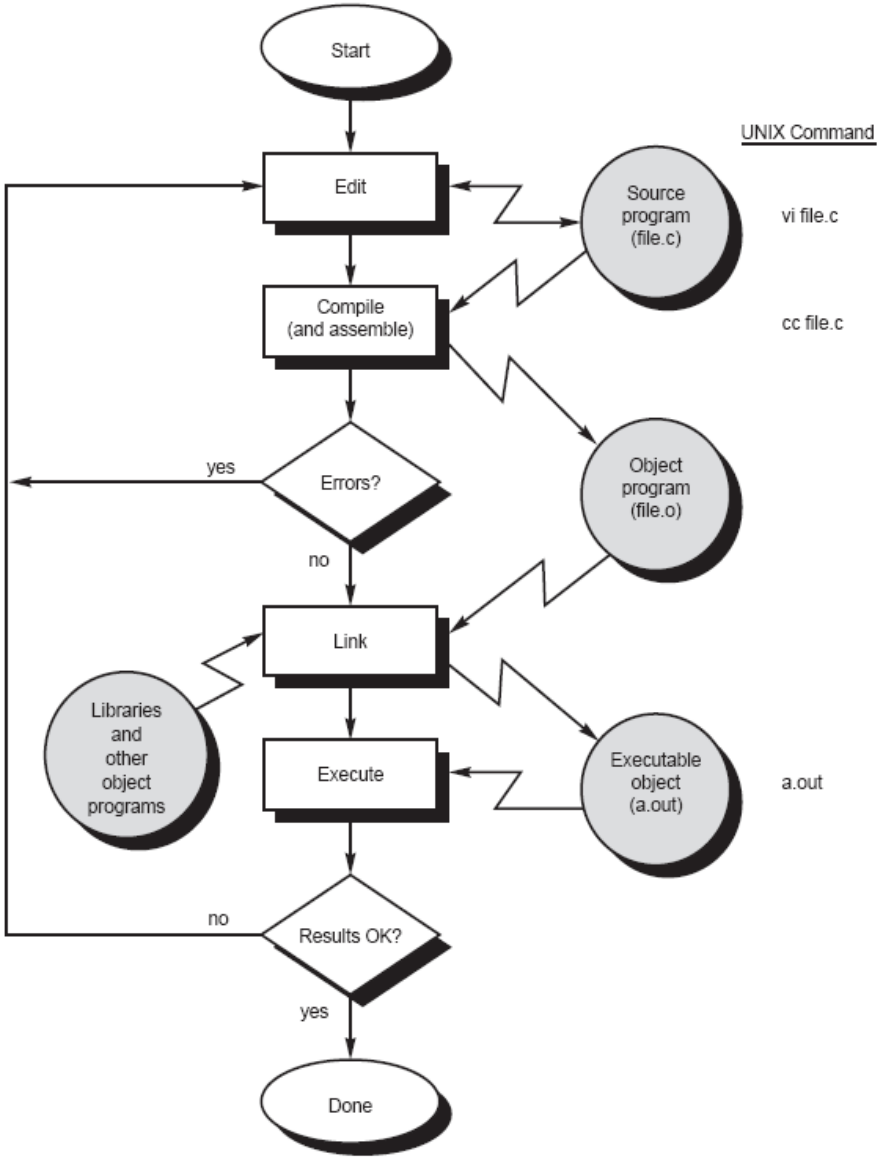
```
/* Hello World! */
#include <stdio.h>

int main()
{
    printf("Hello World!\n");
    return 0;
}
```

Walk through:

- C program: hello.c
  - emacs, vi, vim, pico, joe …
  - But text editors only. No word processor
- Use GNU CC
- Preprocessing: hello.s, assembly code
  - gcc -S hello.c
- Compilation: hello.o, a binary file
  - gcc -c hello.s
- Linking: a.out or hello, an executable file
  - gcc hello.o
  - gcc -o hello hello.o
- Loading (dynamical linking) and execution: ./hello
  - ./a.out
  - ./hello

Or
gcc –g –Wall hello.c –o hello

- A typical execution of a piece of C program



Sample 1 Dissection:

- What it has?
  - A line of comment
  - A line of preprocessor directive
  - A function definition: main
  - An output statement
  - A return clause
- What it does?
  - Ask the computer to say hello to the world.
- What it does not do?
  - It seems not computing!!
  - No real work
    - not taking input
    - does not change any value

```
/* Hello World! */
#include <stdio.h>

int main()
{
    printf("Hello
World!\n");
    return 0;
}
```

Sample 2:

```
#include <stdio.h>
#define MAGIC 10
int main()
{
    int i=0, fact, quotient;
    while (i++ < 3) {
        printf("Guess a factor of MAGIC larger than 1: ");
        scanf("%d", &fact);
        quotient = MAGIC % fact;
        if (0 == quotient)
            printf("You got it!\n");
        else
            printf("Sorry, You missed it!\n");
    }
    return 0;
}
```

Sample 2 Dissection:

- What more it has?
  - Macro definition
  - Variable declaration
  - Operations represented by operators
  - Conditional computation
  - Input Processing
  - Loops: three chances