CS2311 Computer Programming

About the Course

Y MONG

ASSOCIATE PROFESSOR

DEPT. OF COMPUTER SCIENCE

COURSE SITE

About the Course

- Teaching pattern
  - Lectures
    - Explain the terminologies, concepts, methodologies, …
  - Labs
    - Hands-on programming practice
    - Analyzing example problems and implementing programs

- Canvas-based course website
  - Teaching materials are all in Canvas
  - It is your own responsibility to check Canvas and University emails regularly for updates

About the Course

- Assessment
  - Coursework (40%)
    - Assignments (15%)
    - Midterm quiz (15%)
    - Lab Exercises (10%)
      - Randomly choose 8-10 labs to assess
      - Deadline is 24 hours after your lab session
  - Final examination (60%)

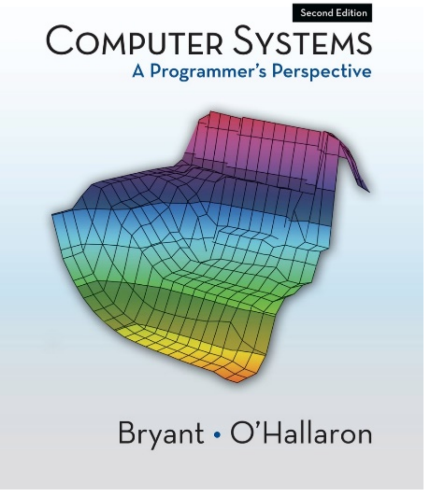
About the Course

- Assessment
  - To **pass** the course you must obtain:
    - At least 30% of the final exam marks (No. 1 reason to fail this course)

Student	Coursework	Exam	Final Mark	Grade
1	94.3	95.5	95.14	A+
2	33.8	34	34	D
3	86.8	26.5	44.59	F

About the Course

- Resources
  - Textbook (NIL)
  - Reference books
    - Computer Systems: A Programmer‛s Perspective, by Randal E. Bryant and David R. O'Hallaron, Prentice Hall, 2011



  - Microsoft Visual Studio 2019 (Windows)
    - Develop environment for compiling & debugging
  - PASS (Program Assignment aSsessment System)
    - Program testing and submission

About the Course

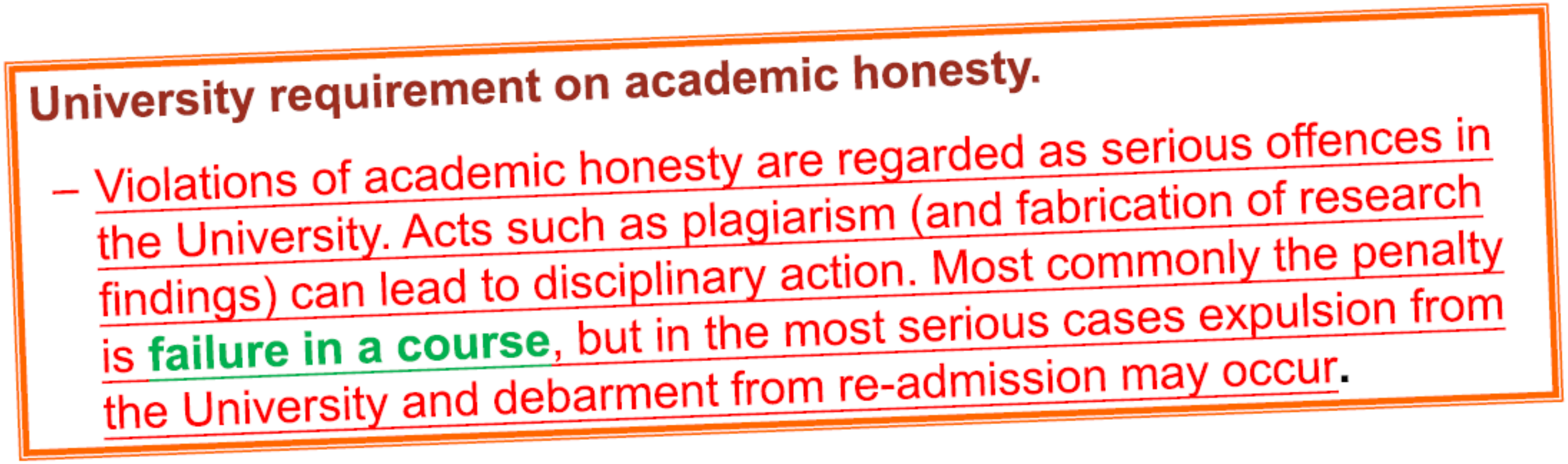
- Key to success

Just Do It

But, do it yourself

About the Course

- "Do it yourself" means
  - Discuss the problems with any other people
  - Study materials on the internet
  - Refer to any books
- **But**, the details and write-ups must be entirely your own work



University requirement on academic honesty.

– Violations of academic honesty are regarded as serious offences in the University. Acts such as plagiarism (and fabrication of research findings) can lead to disciplinary action. Most commonly the penalty is **failure in a course**, but in the most serious cases expulsion from the University and debarment from re-admission may occur.

About the Course

- Things draw your attentions
  - Plagiarism
    - Punishment ranges from warning to course failure
    - May cause you be forced out of CityU
    - Can be automatically detected by PASS system
  - How to prevent
    - In plagiarism cases, both the **giver** and **copier** get punishments
    - Protect your code
  - As instructors
    - We have the responsibility to report academic dishonesty cases so as not to compromise the quality of education
    - We take suspected plagiarism cases very seriously.

About the Course

- How to get an **A**
  - Studying
    - Be prepared to the class (try to read the materials before hand)
    - Attend and participate in all classes (lec & lab)
  - Practice
    - Do labs yourself and repeat the practice
    - Don’t be afraid of asking questions
  - Assignments
    - Start as early as possible and submit on time
    - Many times, debugging takes much longer time than you expect, so make a good plan

Outline for Today

- What is a Computer?
- What is a Computer Program?
- Programming Language
- Programmer
- Basic Concept of programming
- Simple Program

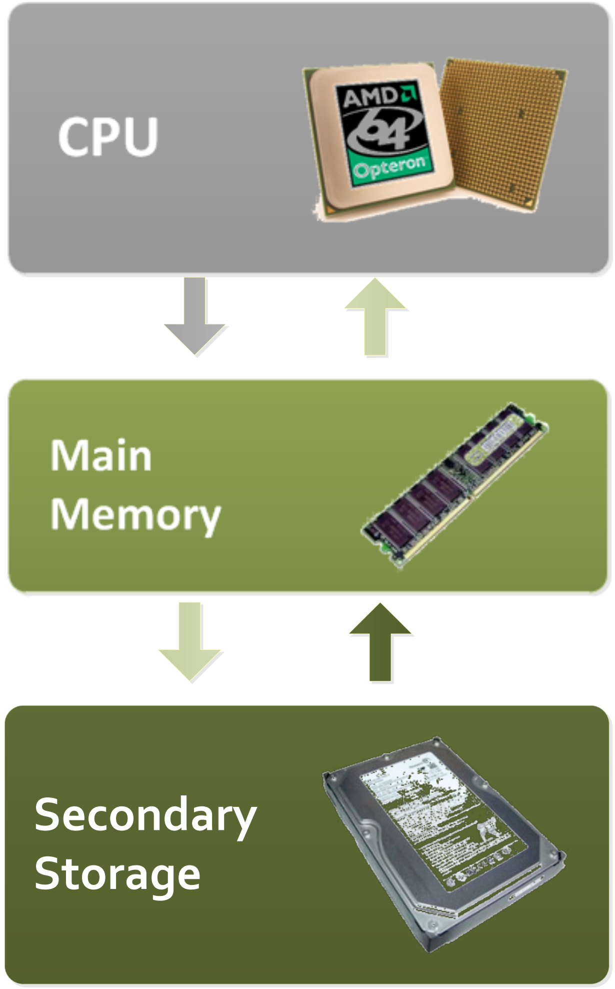
Why Learn Programming

Why Learn Programming

"Everybody in this country should learn how to program a computer, because it teaches you how to think."

Critical thinking: Logic Rigor Clarity

Personal Computer



CPU (Central Processing Unit): Read instruction from main memory and execute the instruction. Update main memory value or send instruction to motherboard

Main Memory: fast storage of program and data in action

Secondary Storage: (slow) Storage of program and data files (e.g., maintain them after shutting down the computer)

Personal Computer

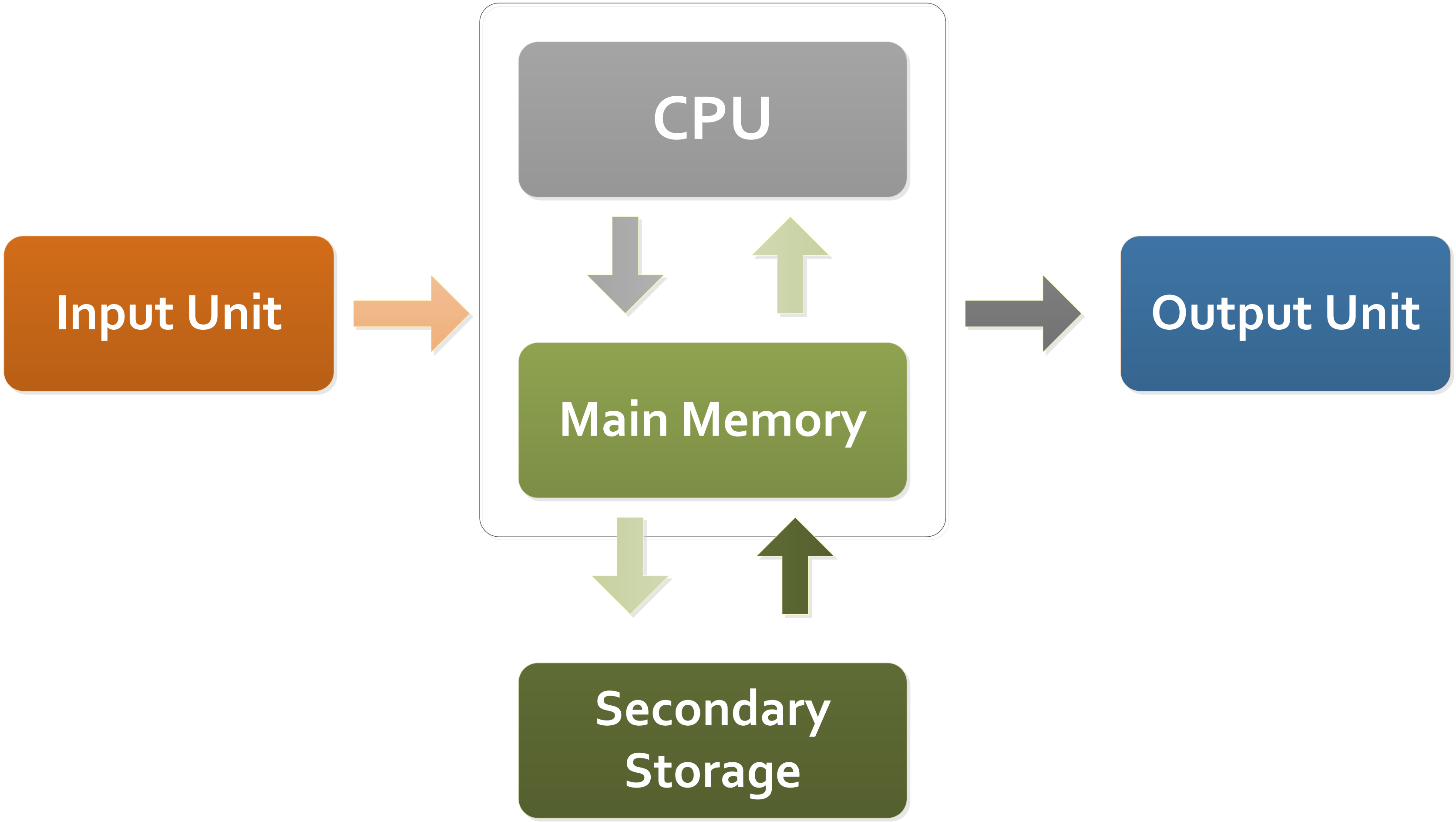
Input Unit

Input Unit: Get input from user or external environment

Output Unit

Output Unit: Show result to user or other programs

Stored Program Computer (Von Neumann Machines)



CPU

Input Unit

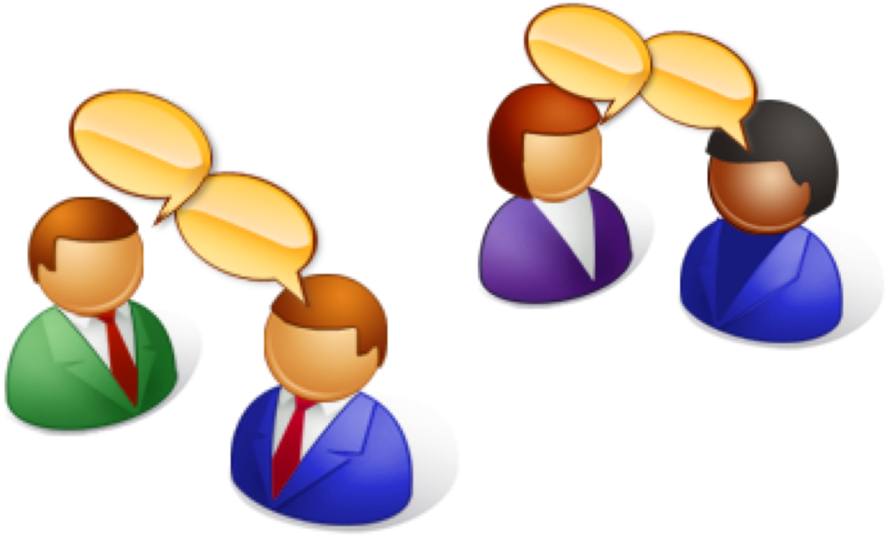
Main Memory

Output Unit

Secondary Storage

What is a Computer Program?

- A way to communicate with computers
- Written in *Programming Languages*

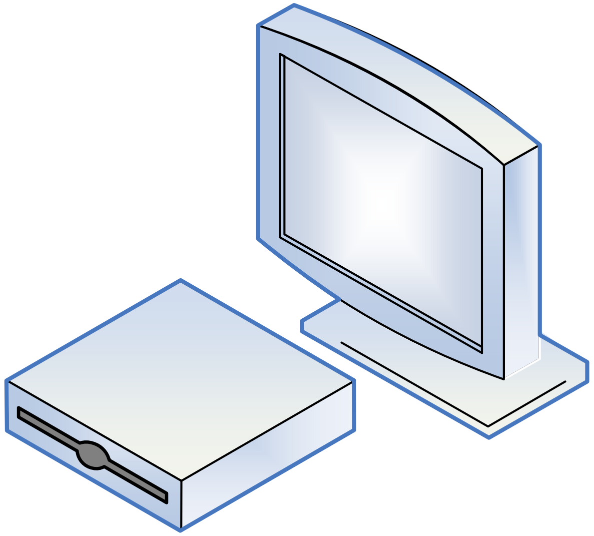


What is a Computer Program?

- A list of instructions that tells a computer to do something

Timer Recording

1. Turn on
2. Set Channel to **ch01**
3. Set Date to **6/9/2019**
4. Set Time to **3:00pm**
5. Confirm setting



Program X

```
int x=10;
int y=11;
y+=x;
System.out.println(y);
System.out.println(x);
```

Programming Languages

- To write a program for a computer, we must use a computer language.

Machine Language

Directly understood
by the computer

binary code

PROGRAM 1-1 The Multiplication Program in Machine Language

```
         00000000 00000100 0000000000000000
01011110 00001100 11000010 0000000000000010
         11101111 00010110 0000000000000101
         11101111 10011110 0000000000001011
11111000 10101101 11011111 0000000000010010
         01100010 11011111 0000000000010101
11101111 00000010 11111011 0000000000010111
11110100 10101101 11011111 0000000000011110
00000011 10100010 11011111 0000000000100001
11101111 00000010 11111011 0000000000100100
01111110 11110100 10101101
11111000 10101110 11000101 0000000000101011
00000110 10100010 11111011 0000000000110001
11101111 00000010 11111011 0000000000110100
         01010000 11010100 0000000000111011
                  00000100 0000000000111101
```

The only language understood by computer hardware is machine language.

Programming Languages

- To write a program for a computer, we must use a computer language.

Machine Language

Directly understood by the computer

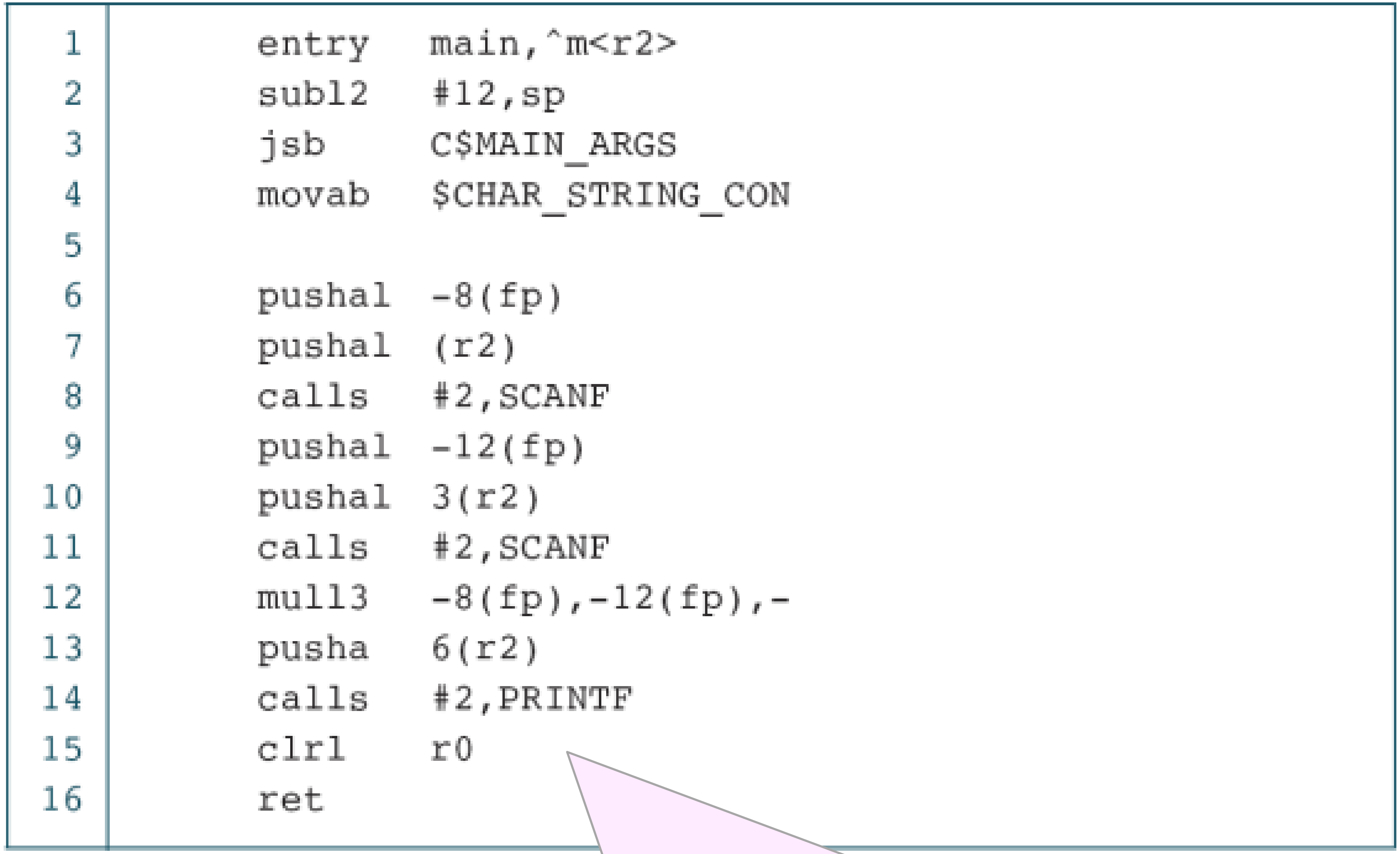
Symbolic Language

English-like abbreviations representing elementary computer operations

binary code

assembly language

PROGRAM 1-2 The Multiplication Program in Symbolic Language



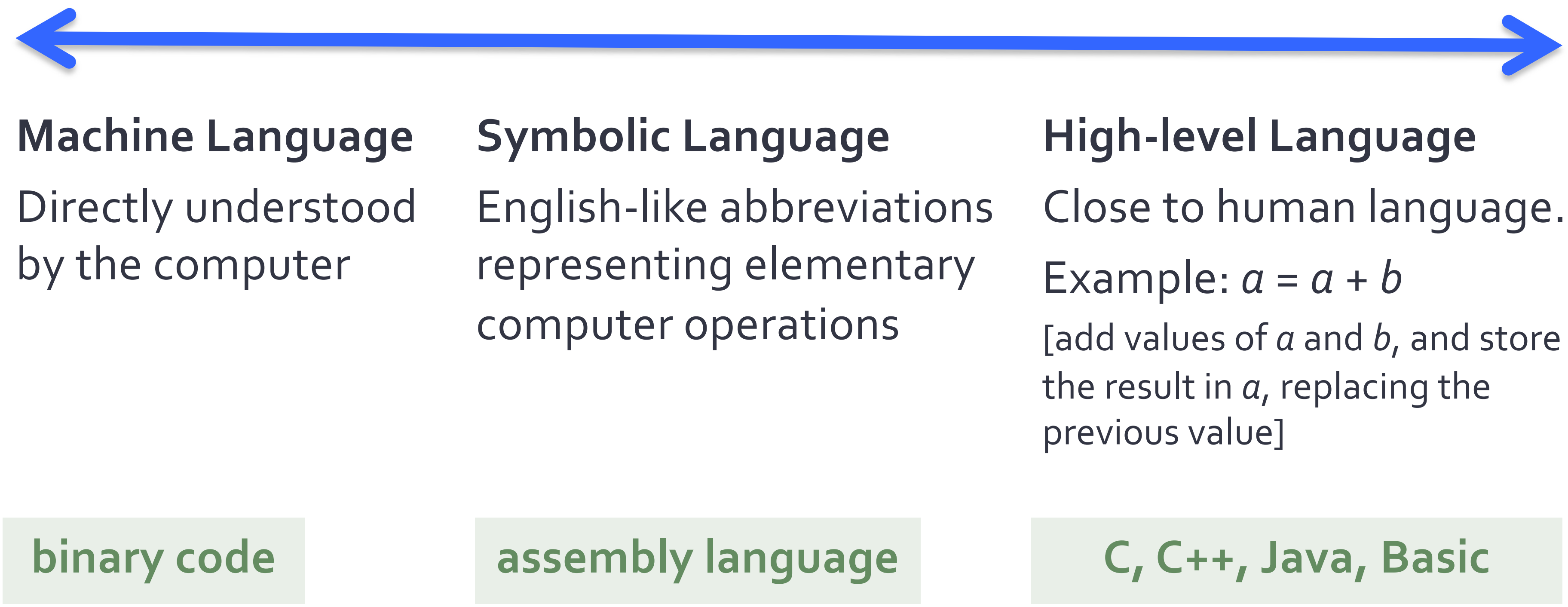
```
        entry   main,^m<r2>
        subl2   #12,sp
        jsb     C$MAIN_ARGS
        movab   $CHAR_STRING_CON

        pushal  -8(fp)
        pushal  (r2)
        calls   #2,SCANF
        pushal  -12(fp)
        pushal  3(r2)
        calls   #2,SCANF
        mull3   -8(fp),-12(fp),-
        pusha   6(r2)
        calls   #2,PRINTF
        clrl    r0
        ret
```

Symbolic language uses symbols, or mnemonics, to represent the various machine language instructions.

Programming Languages

- To write a program for a computer, we must use a computer language.



Machine Language

Directly understood by the computer

binary code

Symbolic Language

English-like abbreviations representing elementary computer operations

assembly language

High-level Language

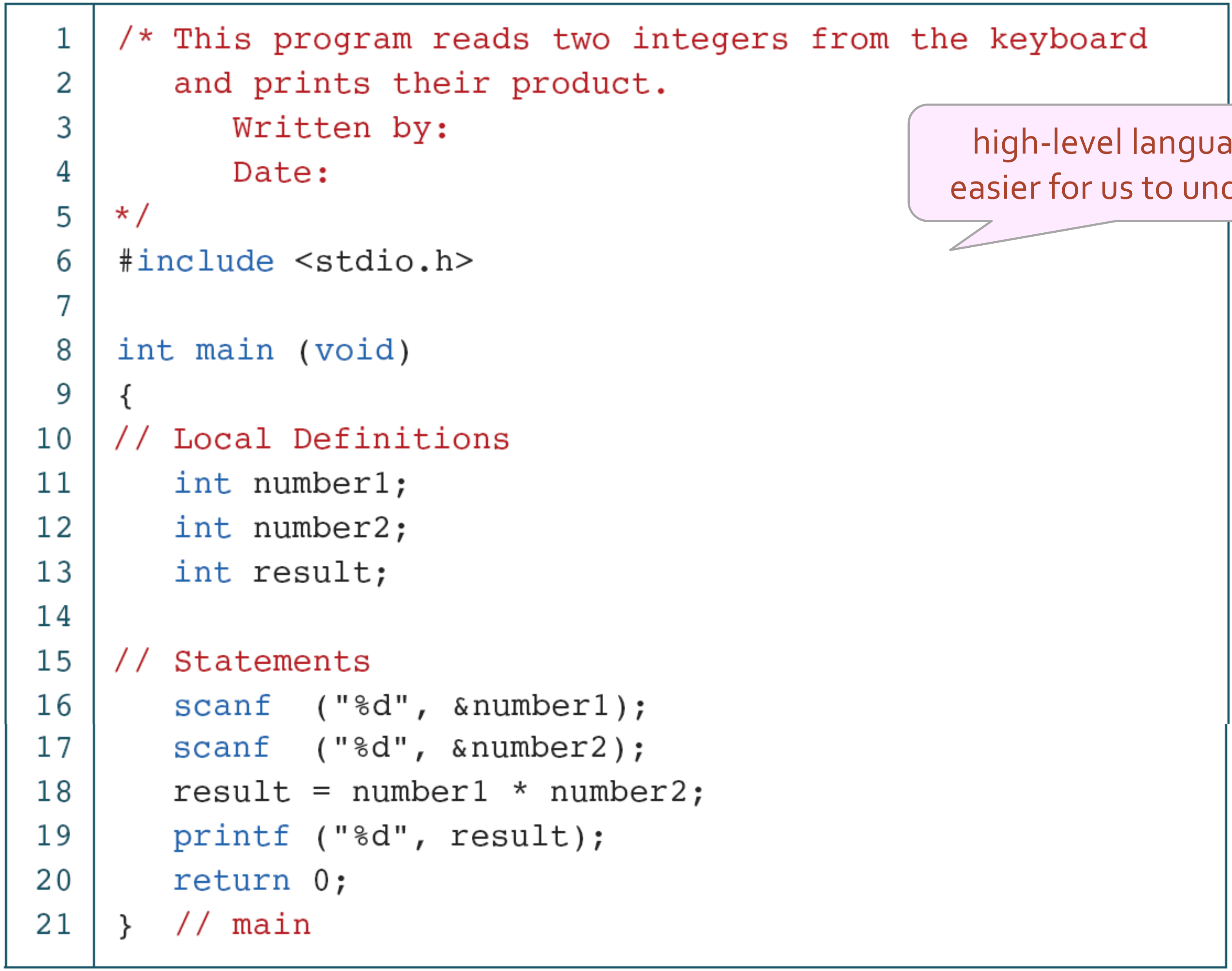
Close to human language.

Example: $a = a + b$

[add values of *a* and *b*, and store the result in *a*, replacing the previous value]

C, C++, Java, Basic

PROGRAM 1-3 The Multiplication Program in C



```c
/* This program reads two integers from the keyboard
   and prints their product.
      Written by:
      Date:
*/
#include <stdio.h>

int main (void)
{
// Local Definitions
   int number1;
   int number2;
   int result;

// Statements
   scanf  ("%d", &number1);
   scanf  ("%d", &number2);
   result = number1 * number2;
   printf ("%d", result);
   return 0;
}  // main
```

high-level languages are easier for us to understand.

There are Many Programming Languages in the World

Ada Assembly Basic C C++ C# Cobol Cobra CODE ColdFusion Delphi Eiffel Fortran FoxPro GPSS J# J++ Java JavaScript LISP Logo LUA MEL Modula-2 Miranda Objective-C Perl PHP Prolog Python SQL Visual Basic Swift

Programming Languages

- Programming languages usually differ in two aspects
  - Language Syntax
  - Standard libraries / SDKs / functions
- Java

```
if (a>b) {
    System.out.println("a is larger than b");
} else {
   System.out.println("a is smaller than or equal to b");
}
```

- Pascal

```
if a>b then
   writeln('a is larger than b');
else
   writeln('a is smaller than or equal to b');
```

Programming Languages

- Syntax is **well-defined**, NO exception
  - if (…){…} else {…};
  - for (;;;) {…}
  - while () {…}

- Basic Components:
  - Variables / structures / function declaration
  - Variables / structures / function access
  - Conditional statements
  - Iteration statements
  - SDK/built-in functions