

計算機程式設計

Introduction to Computer Programming

# Lecture01: Introduction and Hello World

2/18/2013

Slides modified from Yin Lou, Cornell CS2022: Introduction to C

1

---

---

---

---

---

---

---

---

## What is Computer Programming?

- Give (lots of) simple instructions that tell the computer what to do.
- Combining these **simple instructions** is a computer program.
- What are these instructions written in?
  - A computer language
  - C, C++, C#, Java, Python, ...

2

---

---

---

---

---

---

---

---

## Endless examples of Computer Programs



3

---

---

---

---

---

---

---

---

# Why Learn Programming?

---

---

---

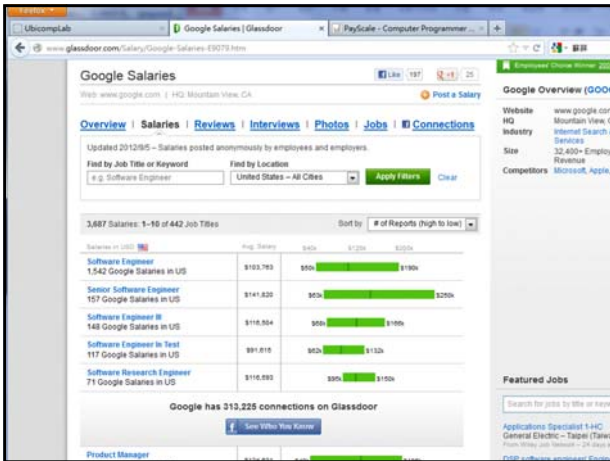
---

---

---

---

---



---

---

---

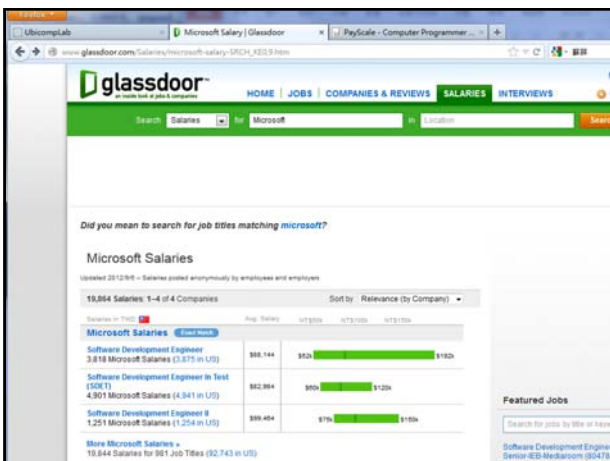
---

---

---

---

---



---

---

---

---

---

---

---

---



### Administrative things

- Instructor: Hao Chu, Room 518
  - Email: [hchu@csie.ntu.edu.tw](mailto:hchu@csie.ntu.edu.tw)
  - Weekly office hours: Wed 2-3:30 pm.
- TA: 陳南蓁 (Nan)
  - Email: [nanchen.chen+introc@gmail.com](mailto:nanchen.chen+introc@gmail.com)
  - Office hours: Thu 12:30 – 1:30, Room 217
- Course URL
  - [http://mll.csie.ntu.edu.tw/course/comp\\_prog\\_s13/](http://mll.csie.ntu.edu.tw/course/comp_prog_s13/)

---

---

---

---

---

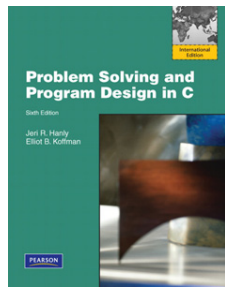
---

---

---

### Textbook (optional)

- C is a simple language, my slides ought to cover it.
  - Print slides (pdf) before classes
- Google to get additional examples.
- Optional textbook: “Problem Solving and Program Design in C,” 6<sup>th</sup> Edition” by Jeri Hanly and Elliot Koffman



---

---

---

---

---

---

---

---

### Goals

- C syntax (vocabulary)
- Standard libraries (reusable code written by others)
- “Structured” programming

---

---

---

---

---

---

---

---

## Grading

- 13 Assignments (33%)
  - Expect 2-10 hours of coding per week
- Midterm exam (33%)
- Final exam (33%)

13

---

---

---

---

---

---

---

---

## 加簽

- Graduate students who must learn programming to do their research
- Students who are not going to drop this course at a later time (particularly due to heavy course load)

14

---

---

---

---

---

---

---

---

## Topics

- |                                |                             |
|--------------------------------|-----------------------------|
| Lect02: C Overview             | Lect09: Preprocessor        |
| Lect03: Control Flow           | Lect10: File Input & Output |
| Lect04: Basic Types & Function | Lect11: Structures          |
| Lect05: Array & Recursion      | Lect12: Linked List         |
| Lect06: Pointer & Strings      | Lect13: Binary Tree         |
| Lect07: Variable Scope         | Lect14: Other C Topics      |
| Lect08: Review                 | Lect15: Programming Style   |

4/15 Midterm exam

6/17 Final exam

15

---

---

---

---

---

---

---

---

## Programming Environment

- Codeblocks
- Use other editors or your choices



---

---

---

---

---

---

---

---

## Ready to write your 1<sup>st</sup> program?

- Get your accounts
- Find the Codeblocks program already on the computer

17

---

---

---

---

---

---

---

---

## CodeBlocks: Compile and Run

- Run CodeBlocks
- Create a new project
  - File → New → Project
  - Select "Console application"
  - Click C → Next
  - Type in Project title: (e.g., "hello")
  - Click finish
- Open and edit the main.c
  - File → Open
  - Find "main.c"
- Compile and run
  - Build → Build and Run

18

---

---

---

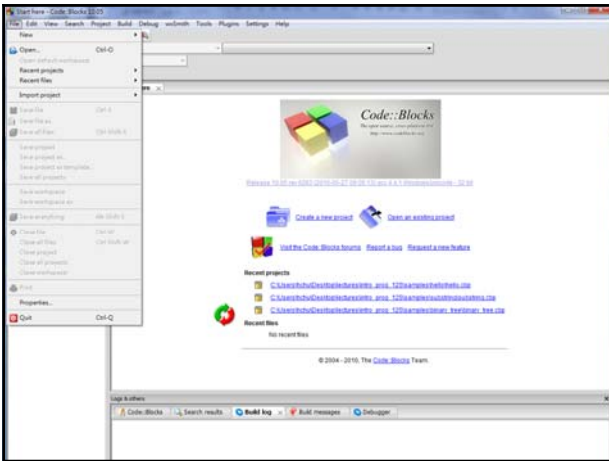
---

---

---

---

---



---

---

---

---

---

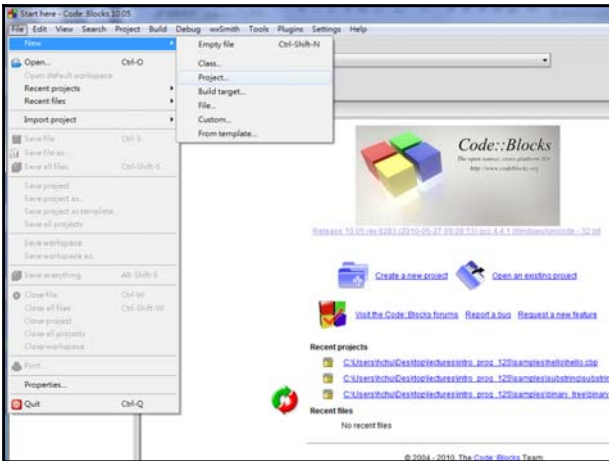
---

---

---

---

---



---

---

---

---

---

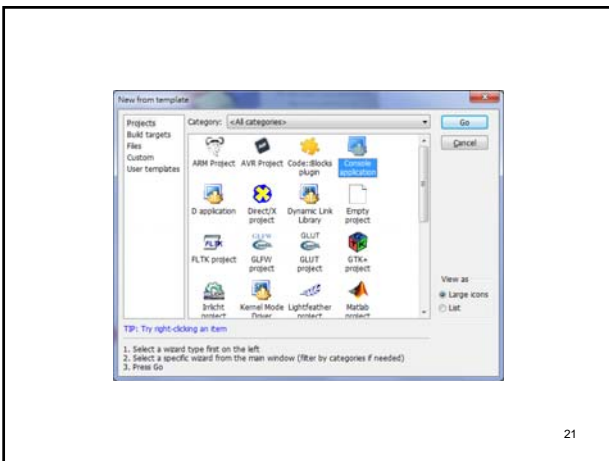
---

---

---

---

---



---

---

---

---

---

---

---

---

---

---



22

---

---

---

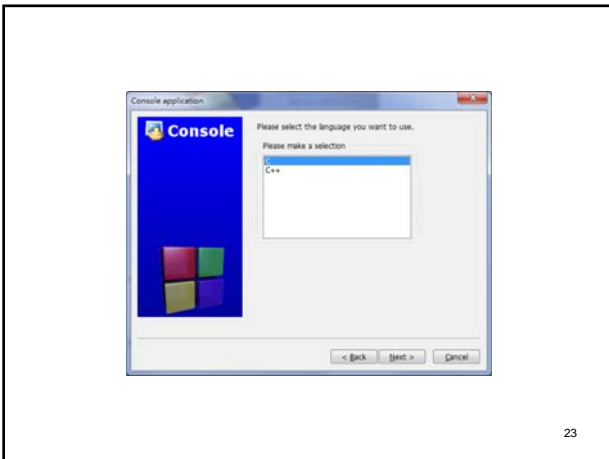
---

---

---

---

---



23

---

---

---

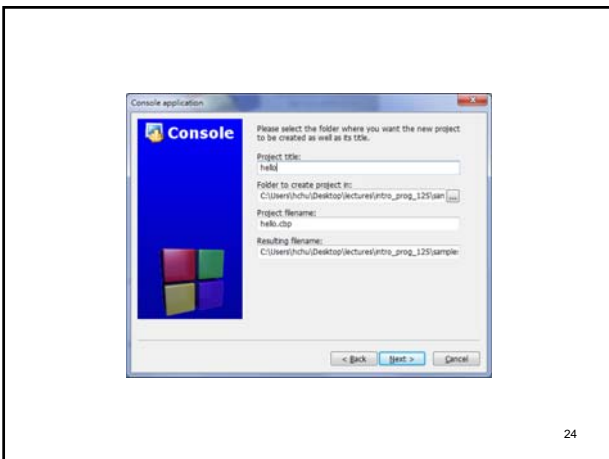
---

---

---

---

---



24

---

---

---

---

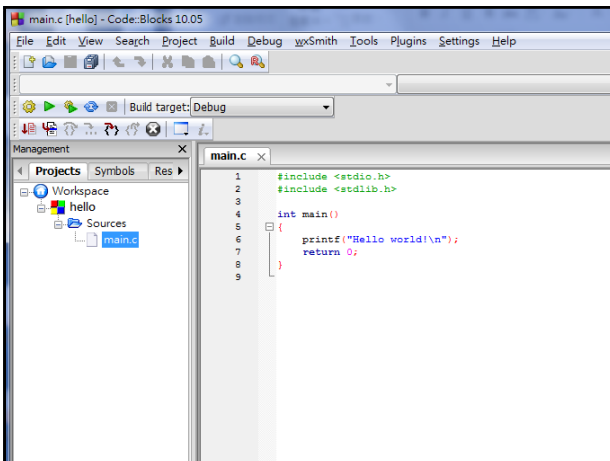
---

---

---

---





---

---

---

---

---

---

---

---

---

---

**main.c: Hello World** (delete the original program)

```
#include <stdio.h>

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

26

---

---

---

---

---

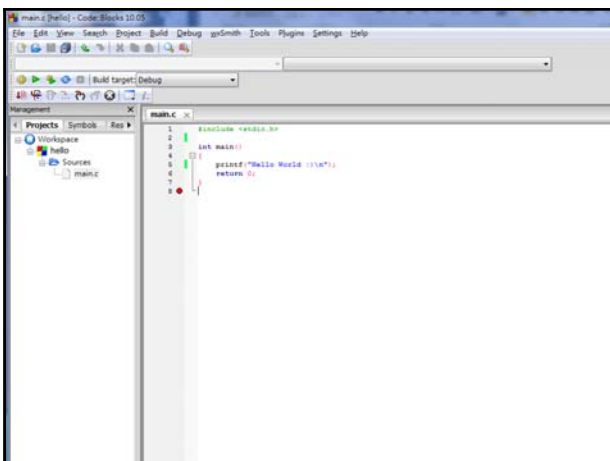
---

---

---

---

---



---

---

---

---

---

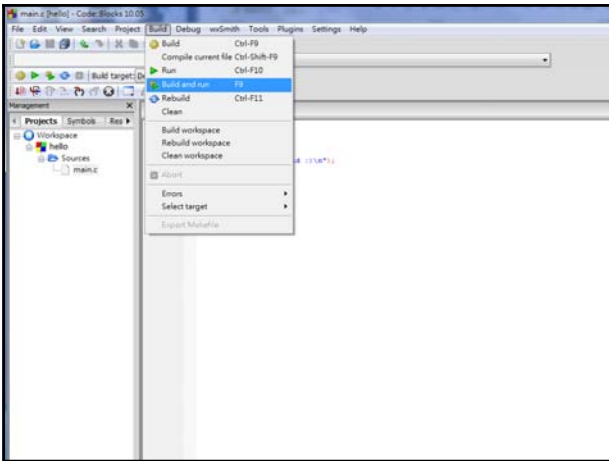
---

---

---

---

---




---

---

---

---

---

---

---

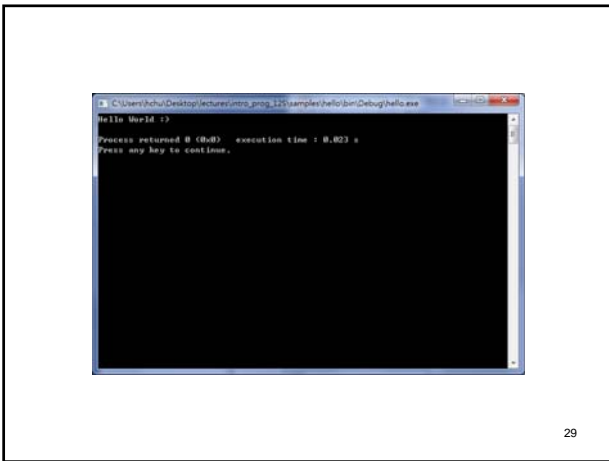
---

---

---

---

---




---

---

---

---

---

---

---

---

---

---

---

---

### What Happened?

- Compile (Build → Build)
  - Compile “main.c” to machine code named “hello.exe”
- Run (Build → Run)
  - Execute the program “hello.exe”

```

main.c (Hello World)

#include <stdio.h> /* printf() is declared in this header file. */

int main() /* Main point of execution */
{
    printf("Hello World :)\n"); /* Output "Hello World" to console */
    return 0; /* Tell OS the program terminates normally */
}
    
```

---

---

---

---

---

---

---

---

---

---

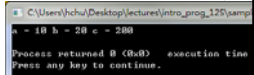
---

---

### main.c: Variables

```
#include <stdio.h>

int main()
{
    int a, b, c;
    a = 10;
    b = 20;
    c = a * b;
    printf("a = %d b = %d c = %d\n", a, b, c);
    return 0;
}
```




---

---

---

---

---

---

---

---

### Review: Hello World

what is #include <stdio.h>? a library of functions  
 what are main() and printf()? called functions  
 function: a set of code enclosed in { ... } that takes some inputs, perform some operation, and produces output;  
 example: rect\_area(height, width) = height \* width  
 more about how to define and write functions next week

```
#include <stdio.h>

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

32

---

---

---

---

---

---

---

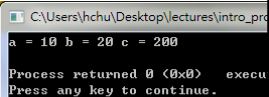
---

### Review: main.c: Variables

a name for a place in memory that stores a value  
 what is computer memory?  
 each variable must have a type  
 C punctuations: ;, {}, (), " "  
 arithmetic (math) operators: + - \* / =

```
#include <stdio.h>

int main()
{
    int a;
    int b, c;
    a = 10;
    b = 20;
    c = a * b;
    printf("a = %d b = %d c = %d\n", a, b, c);
    return 0;
}
```




---

---

---

---

---

---

---

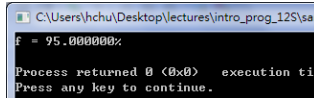
---

## More on printf()

- `printf(format string, arg1, arg2);`
  - **format string** can include **placeholders** that specify how the arguments `arg1`, `arg2`, etc. should be formatted
  - `%c` : format as a character
  - `%d` : format as an integer
  - `%f` : format as a floating-point number (less bits)
  - `%lf` : format as a floating-point number (more bits)
  - `%%` : print a % character

### Examples

```
float f = 0.95;
printf("f = %f%%\n", f * 100);
```




---

---

---

---

---

---

---

---

---

---

---

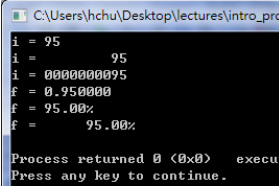
---

## Even more on printf()

- Placeholders can also specify widths and precisions
  - `%10d` : add spaces to take up at least 10 characters
  - `%010d` : add zeros to take up at least 10 characters
  - `%.2f` : print only 2 digits after decimal point
  - `%.5.2f` : print 1 decimal digit, add spaces to take up 5 chars

### Examples

```
int i = 95;
float f = 0.95;
printf("i = %d\n", i);
printf("i = %10d\n", i);
printf("i = %010d\n", i);
printf("f = %f\n", f);
printf("f = %.2f%%\n", f * 100);
printf("f = %10.2f%%\n", f * 100);
```




---

---

---

---

---

---

---

---

---

---

---

---

## Even more on printf() format string

<http://www.cplusplus.com/reference/cstdio/printf/>




---

---

---

---

---

---

---

---

---

---

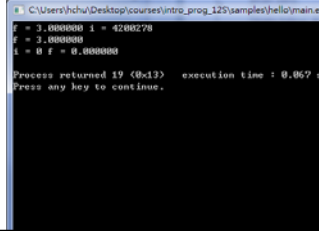
---

---

### Warning about printf()

- printf is powerful, but potentially dangerous
- What does this code output?

```
int i = 90;
float f = 3;
printf("f = %f i = %d\n", f);
printf("f = %f\n", f, i);
printf("i = %d f = %f\n", f, i);
```




---

---

---

---

---

---

---

---

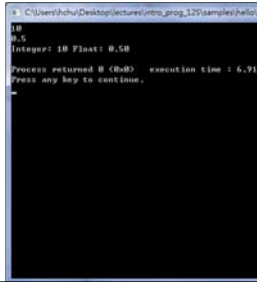
---

---

### main.c: scanf()

```
#include <stdio.h>

int main()
{
    int i;
    double f;
    scanf("%d", &i);
    scanf("%lf", &f);
    printf("Integer: %d Float: %2.2lf\n", i, f);
    return 0;
}
```




---

---

---

---

---

---

---

---

---

---

### In-Class Exercise 1.1

- Write a program that calculates travel reimbursement for a pizza delivery person at a rate of NT\$6.97 per kilometer. Your program should interact with the user in the following manner: (It is okay not to get the program right the first time. Look at the compilation errors and fix them.)

```
KILOMETER REIMBURSEMENT CALCULATOR
Enter beginning odometer readings > 13505.2
Enter ending odometer reading > 13810.6
You traveled 305.4 kilometers. At $6.97 per kilometer,
Your reimbursement is $2128.63
```

---

---

---

---

---

---

---

---

---

---

### In-Class Exercise 1.2

- Write a program that asks for two integers and outputs the results of four arithmetic operations (+, -, \*, /). Below shows the sample output.

```
enter a> 5
enter b> 2
a+b = 7
a-b = 3
a*b = 10
a/b = 2
```

40

---

---

---

---

---

---

---

---

After you are done with in-class exercises, you can start the assignment #1.

TA will talk about how to upload your assignment on the judge system

41

---

---

---

---

---

---

---

---