

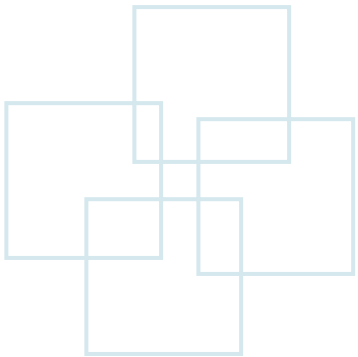
Understand the Basic Structure of Programming Languages

高階語言程式實習

Yuan-Hao Chang (張原豪)

johnsonchang@ntut.edu.tw

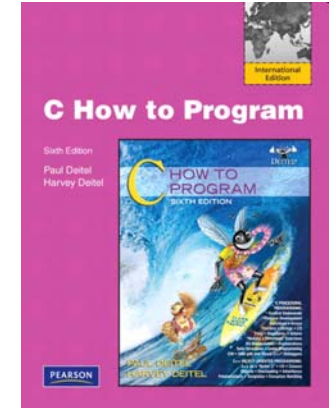
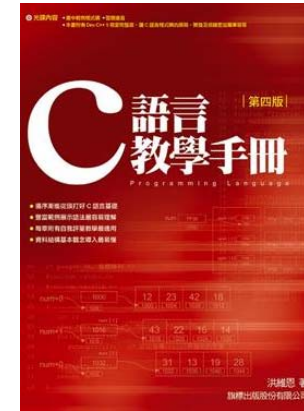
Department of Electronic Engineering
National Taipei University of Technology





Course Information

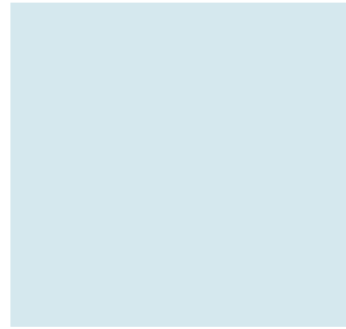
- 授課教師: 張原豪 (207-2 室、分機 2288)
- 上課時間: 星期四 上午 9:10 – 中午 12:00
- 教室: 共同科館 313
- 參考書目:
 - C 語言教學手冊第四版-洪維恩-旗標 (ISBN: 9574424847)
 - 最新C程式語言教學範本 第四版 - 蔡明志
 - C How to Program (5th Edition) by Paul Deitel
 - C++ Primer (4th Edition) by Stanley B. Lippman
 - C Programming Language (2nd Edition) by Brian W. Kernighan and Dennis M. Ritchie
- 課程網頁:
 - <http://www.ntut.edu.tw/~johnsonchang/courses/CLanguage201008/>
- 成績評量: (subject to changes)
 - 作業: (30%), 期中考(30%), 期末考(30%), 平時表現(10%)



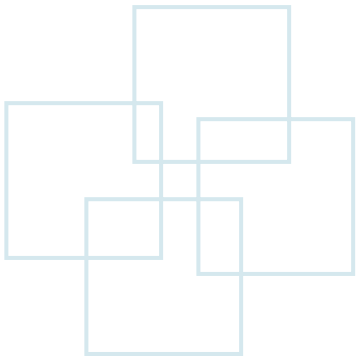


Objectives

- Introduce the basic structure of programming language using C.
 - Get students new to programming language started as soon as possible.
- Provide training in developing programs for solving various scientific problems.
 - Construct C programs.
 - Compile and debug the programs.
 - Run the executable code.



Syllabus





Outline of the Course (1/6)

- Introduction to programming languages.
 - Various programming languages as well as their properties.
 - The history, the structure and advantages of C language.
 - The first simple C program.
- Overview of C
 - The structure of a program
 - Keywords
 - Debugging tools



Outline of the Course (2/6)

- Data types
 - Variables and constants
 - Data types (int, char, float, etc.)
 - Size of data types
 - Data type transformation
- Operators and expressions
 - Logical operator
 - Arithmetic operators
 - Priority
 - Expression



Outline of the Course (3/6)

- Basic input and output
 - printf(), scanf()
 - getchar(), putchar()
- Flow control
 - if-else
 - switch
 - for
 - while
 - do while
 - break and continue



Outline of the Course (4/6)

- Function
 - Declaration of a function
 - Function arguments
 - Variable scope
 - Comparison between functions and preprocessor macros
- Arrays
 - 1-D array
 - 2-D array
 - Passing arrays to functions
 - Sorting arrays



Outline of the Course (5/6)

- Pointers

- Pointer operator
- Pointers and address
- Pointers and functions
- Pointers and arrays

- String

- Strings and characters
- String handling
- Functions related to string



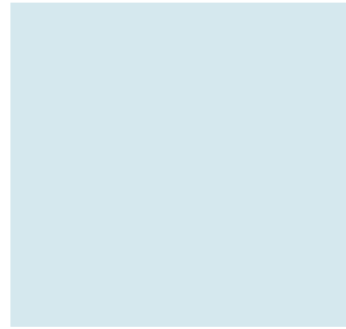
Outline of the Course (6/6)

- Structure
 - Structure definition
 - Structure declaration
 - Nested structure
 - typedef
 - Union
- File I/O
 - File and streams
 - Create a file
 - Read/Write a file



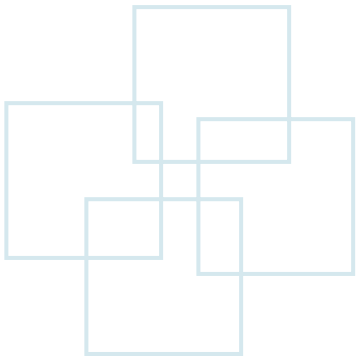
Integrated Development Environment (IDE)

- [Optional] Visual C++ Express Edition
- [Required] Bloodshed Dev-c++
- [Optional] Eclipse IDE for C/C++ Developers
- [Optional] Vim + g++/gcc + gdb (Unix)



Chapter 1

Introduction





History

- B Language was its predecessor
- Designed by Dr. Dennis Ritchie in Bell Lab, 1972
- In 1973, C programming language was used to develop UNIX
- In 1989, ANSI establishes a standard specification of C, called ANSI C



What Is Computer?

- Computer
 - Devices capable of performing computations and making logical decisions
- Hardware
 - Physical components of a computer
 - E.g., Keyboard, mouse, screen, disk, memory, CPU, CD-ROM
- Software
 - Programs that run on a computer



High-level Language

- High-level language
 - Abstract from the details of the computer
 - Portable across various platforms
 - Requiring compiler or interpreter
 - Example: C, C++, Java, Perl, Python
- Low-level language
 - Closer to the hardware
 - Provide the fine-grained control of every function of the machine
 - Example: Assembly language



Compiler and Interpreter

• Interpreter

- Translate each program each line at a time before running it.
- Terminate the program when the interpreter discovers an error.
- Example: Perl, Python, and tcl

• Compiler

- Translate source code from high-level language to low-level language (object code or machine code).
- Create an executable program.
- Example: C, C++, Java



Advantages of C language

- Portability
 - Easy to port the program to newly developed computers
 - Feasible to re-compile the source code and run the executable on any machine.
- Improvability
 - Easy to customize and improve by any programmer



C Standard Library

- A collection of header files and routines used to implement common operations
 - Example: I/O and string handling
- Advantages
 - Avoiding re-writing the same functions
 - Efficient and portable



Write the First Program

1. Construct the program
2. Compile the program
3. Run the executable program



Construct the Program: Hello World!

Example 1

```
// Hello World

#include <stdio.h>

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

Example 2

```
/* sample-01-1: the first sample */
#include <stdio.h>
#include <stdlib.h>
int main() {
    printf("Hello World!!\n");
    system("pause");
    return 0;
}
```

Comments

Library
declaration

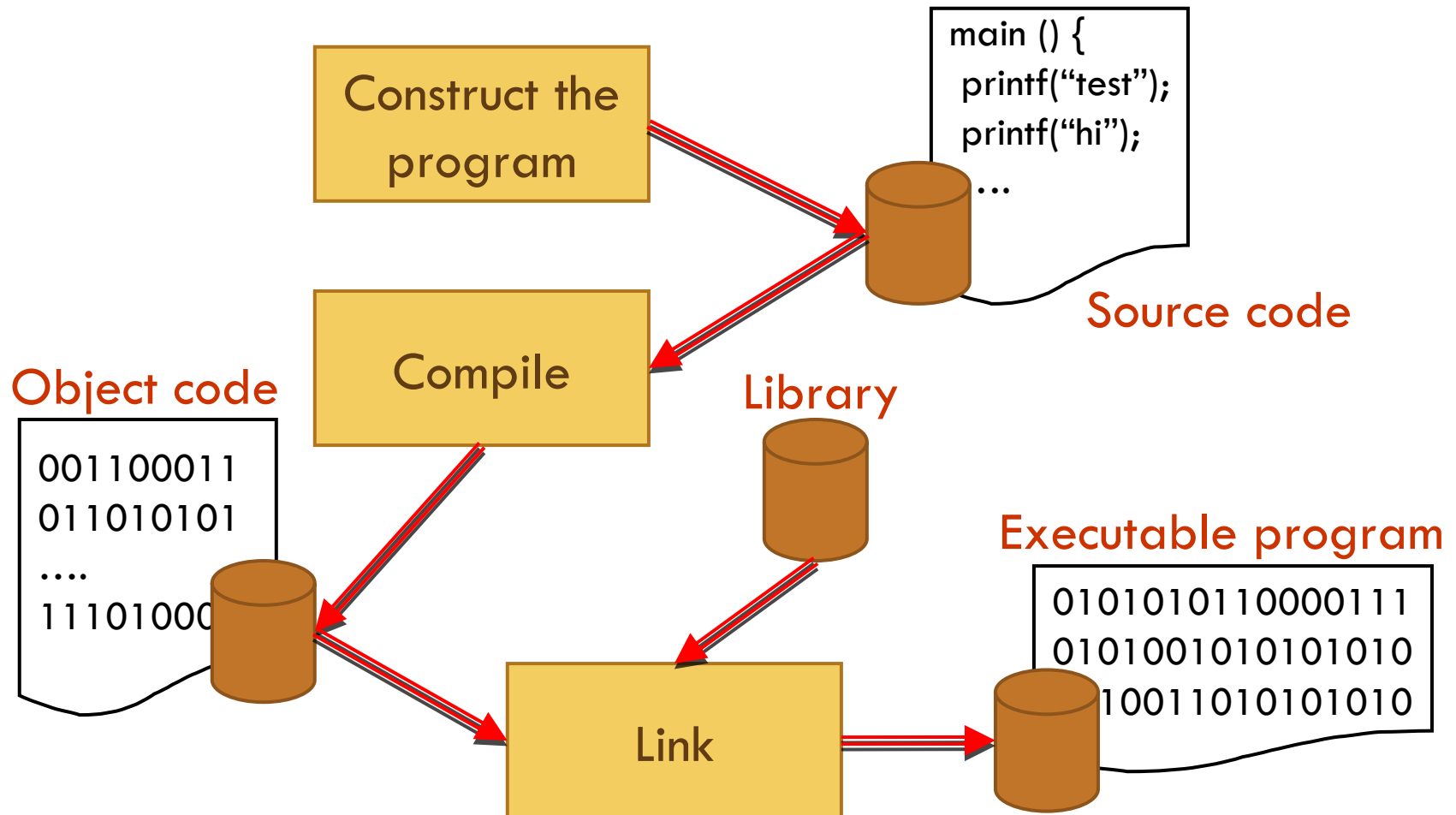
New line

Pause the
program

Print the message
"Hello World!"

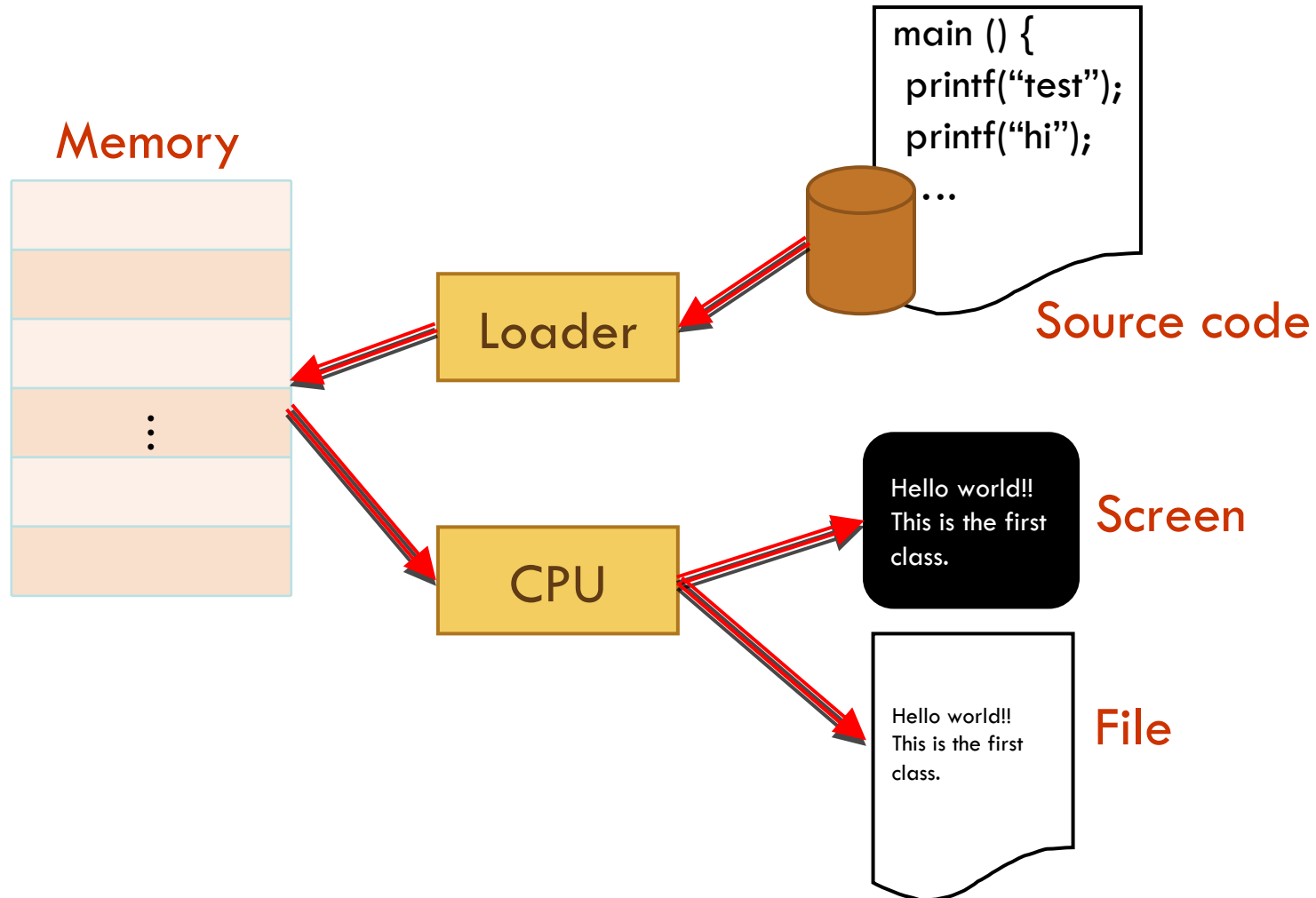


Compile the Program





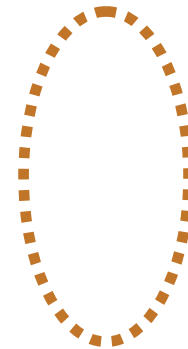
Execute the Program





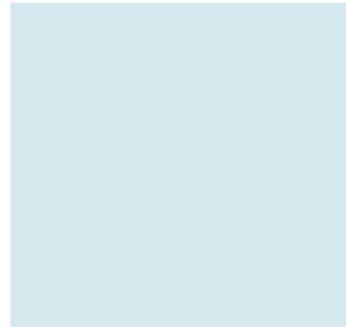
Read Error Messages

```
/* sample-01-1: the first sample */
#include <stdio.h>
#include <stdlib.h>
main() {
    printf("Hello World!!\n");
    system("pause");
}
```

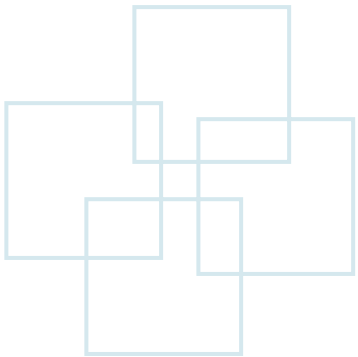


error C2143: 語法錯誤: 遺漏';' (在'}' 之前)

- Double click the error message
- Remove syntax error (語法錯誤)



Bloodshed Dev C++





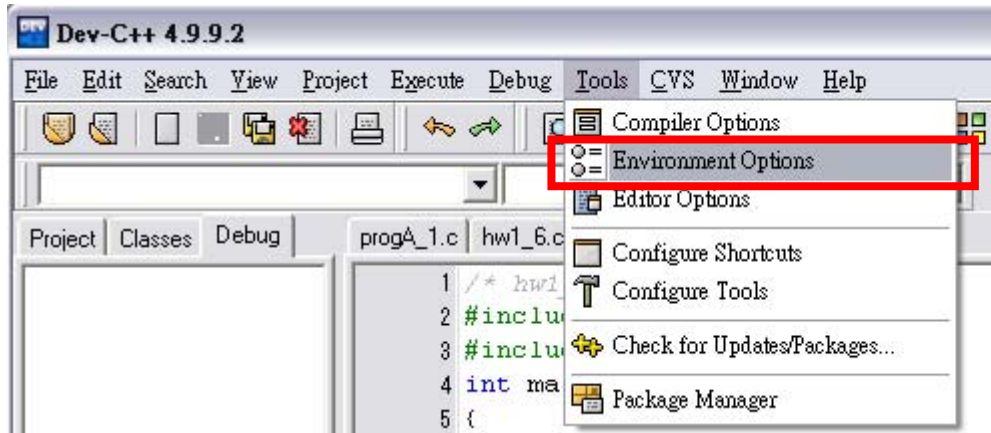
Bloodshed Dev C++

- Dev C++:
 - Provide an IDE for C language by integrating GNU Mingw compiler system with GCC and GDB.
- Designer:
 - Colin Laplace
- Official site:
 - <http://www.bloodshed.net/index.html>
- Installation tutorial:
 - <http://ez2learn.com/index.php/c-tutorials/dev-c-/201-dev-c>
- Download and install
 - Download Dev C++ with Mingw/GCC support:
<http://www.bloodshed.net/dev/devcpp.html>

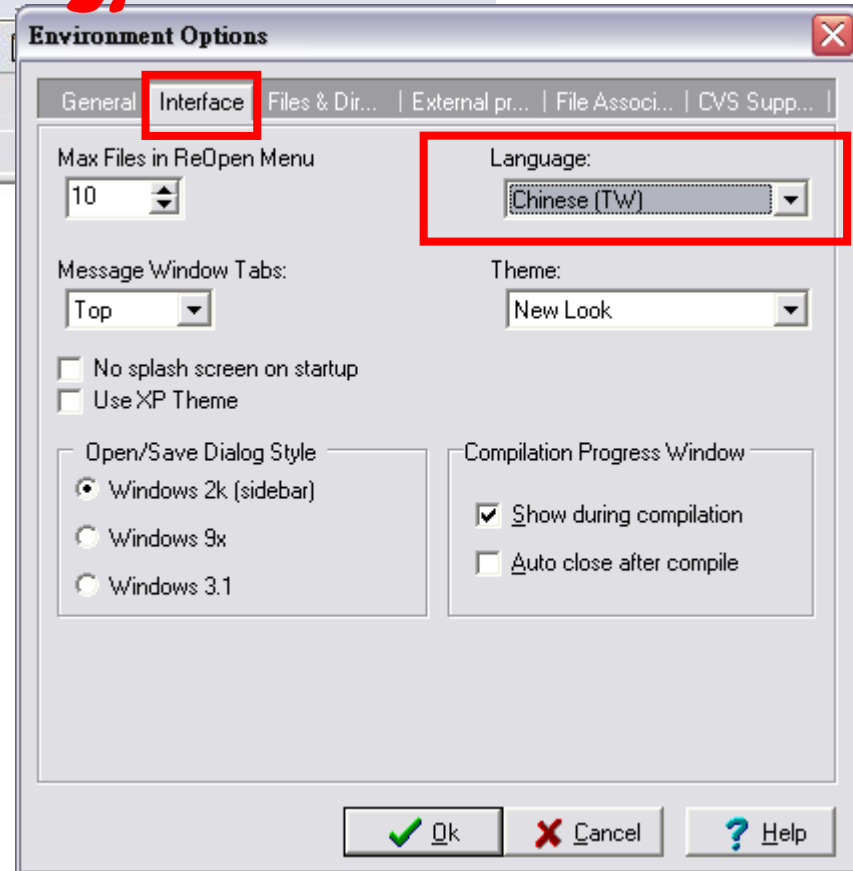


Interface (English / 中文)

1.



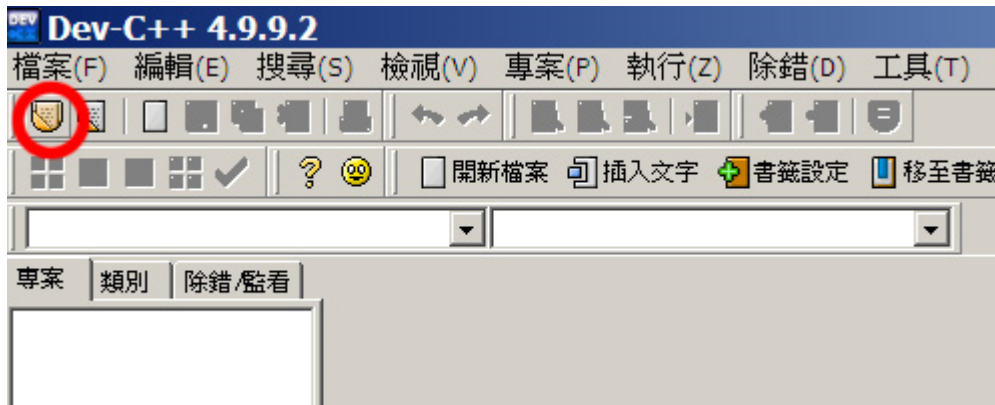
2.



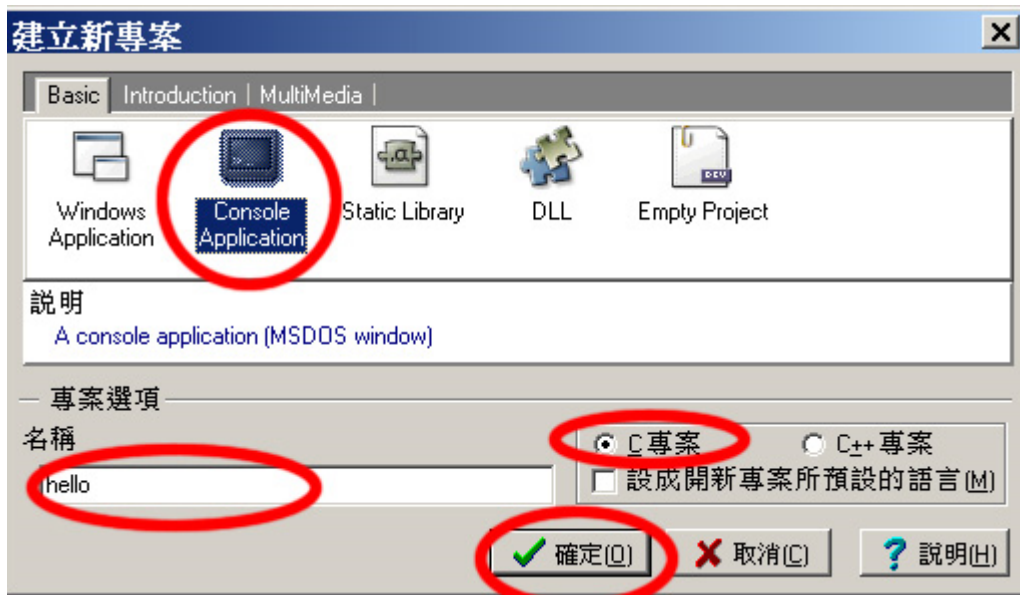


New Project (開新專案)

1.



2.



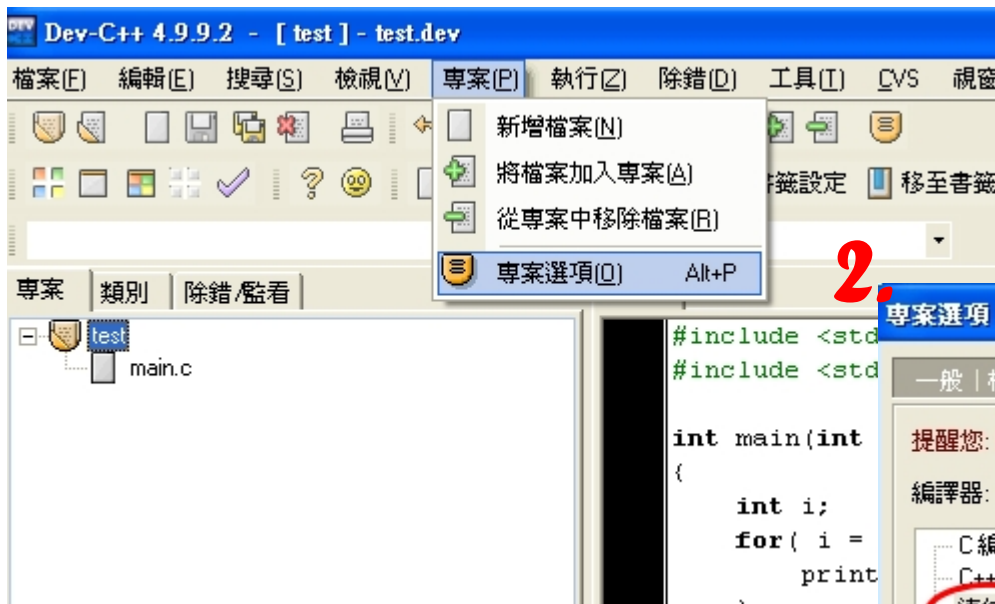
```
#include <stdio.h>
#include <stdlib.h>
```

```
int main(int argc, char *argv[])
{
    printf("Hello World!\n");
    system("PAUSE");
    return 0;
}
```

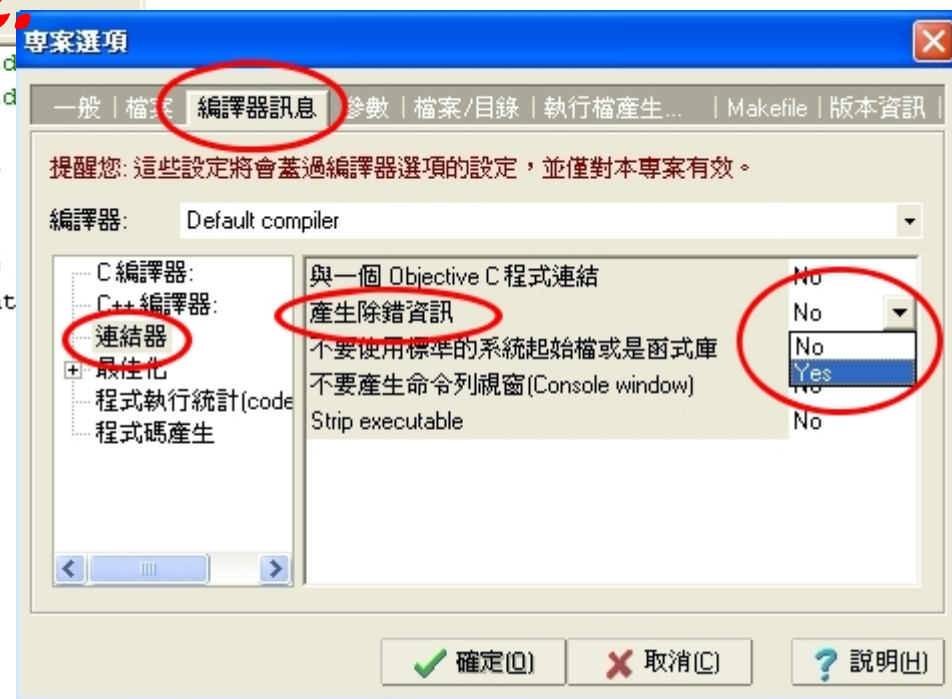


Settings to Enable Debug Information

1.



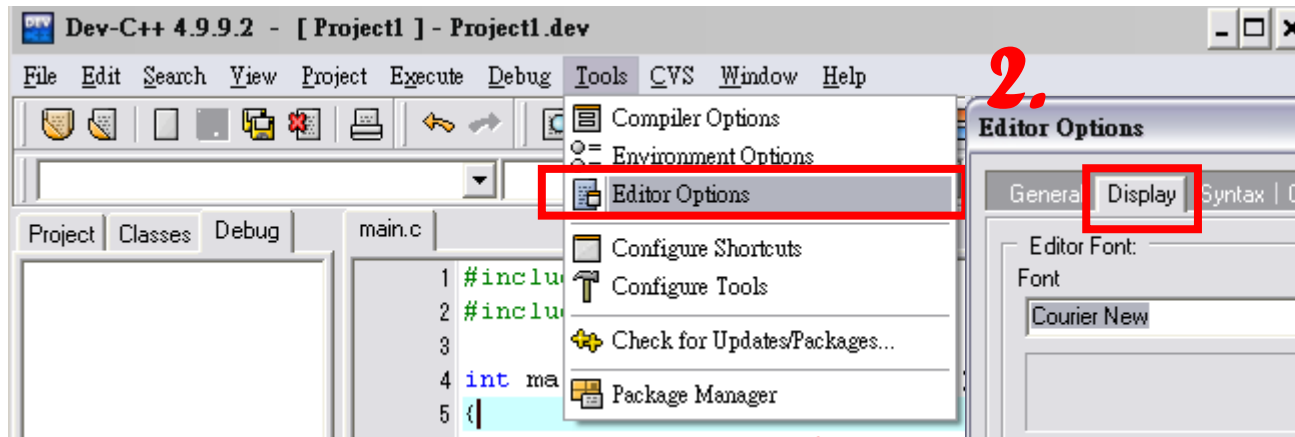
2.



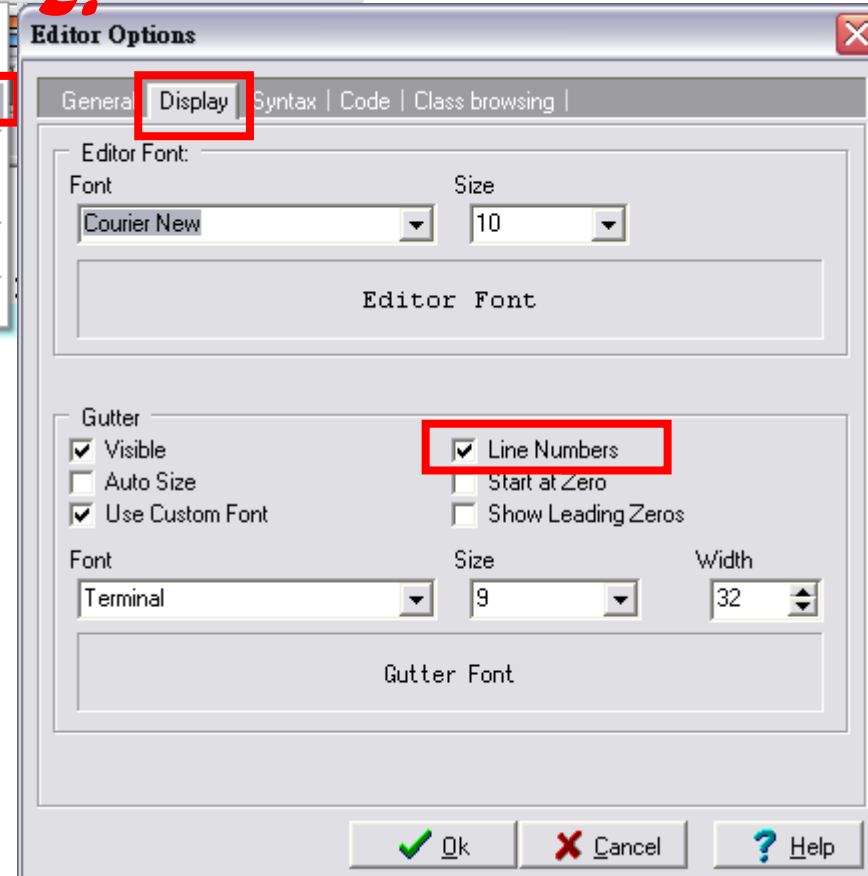


Line Number

1.



2.





Compilation

Compile (Ctrl + F9) Build All (Ctrl + F11)



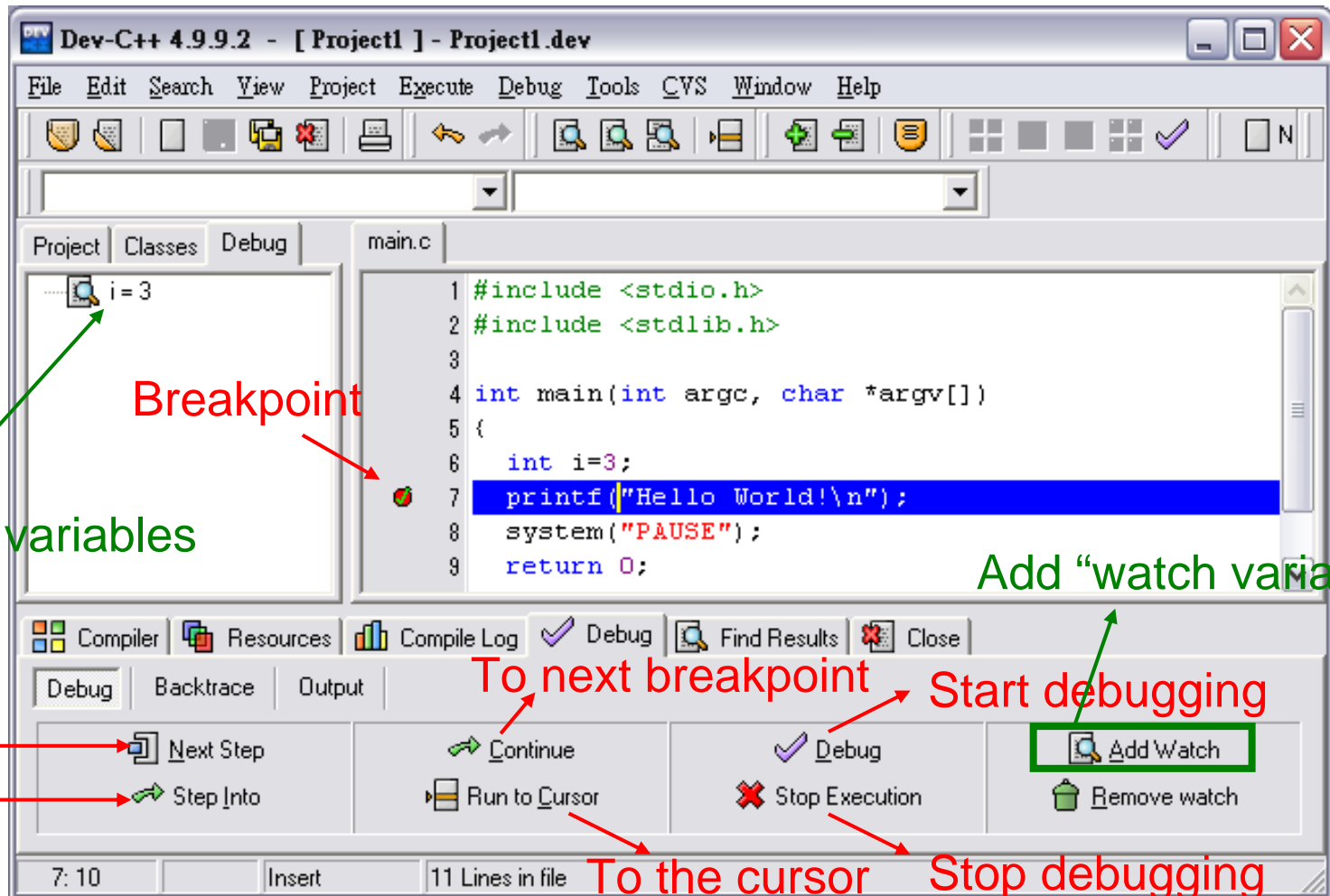
Run (Ctrl + F10)

Compile & Run (F9)

Debug (F8)



Debug



Breakpoint

Watched variables

Add "watch variable"

To next breakpoint

Start debugging

Single step
Single step
into func.

To the cursor

Stop debugging



Lab 01

- Write a program to print the following figure:

```
*  
***  
*****  
*****
```