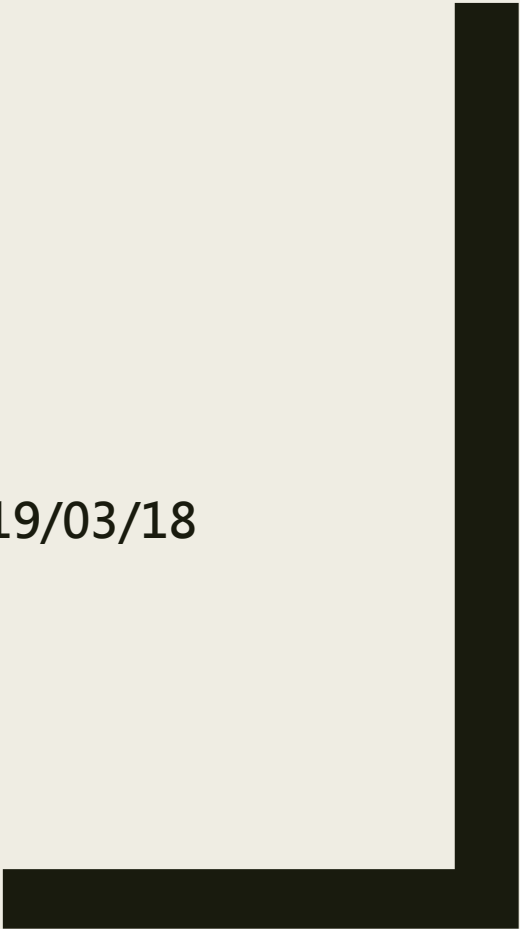




# Lab 2

2019/03/18



# 上機 (1)

## ■ New E3 課程網頁內

The screenshot displays the National Taiwan University Digital Teaching Platform. The top navigation bar is purple with the university's logo and the text '國立交通大學 數位教學平台'. The left sidebar contains several menu categories: '課程資訊' (Course Information) with links to '課程綱要' (Course Syllabus), '成員' (Members), '公告列表' (Announcement List), and '我的郵件' (My Mail); '內容管理' (Content Management) with links to '大綱管理' (Syllabus Management), '教材管理' (Material Management), '作業管理' (Assignment Management), '討論區管理' (Discussion Board Management), '試卷管理' (Exam Management), '題庫維護' (Question Bank Maintenance), and '分組管理' (Group Management); and '評量管理' (Assessment Management) with links to '成績管理' (Grade Management) and '配分設定' (Weight Setting).

The main content area shows the course title '【107下】1190資料結構與物件導向程式設計 Programming'. Below the title, there are two folders: 'Slides 2019/2/18' and 'Slides 2019/3/4'. Further down, under the heading 'TA Courses', there is a folder named '2019/3/4 Practice' which is highlighted with a red rectangle. A green arrow points from the text '考試時間到才會開啟，10分鐘內下載完畢' (The exam time will start, download within 10 minutes) to this folder.

考試時間到才會開啟，  
10分鐘內下載完畢

遲到超過10分鐘，  
該次以0分計

# 上機 (1)

三  國立交通大學 數位教學平台

課程資訊

- 課程綱要
- 成員
- 公告列表
- 我的郵件

內容管理

- 大綱管理
- 教材管理
- 作業管理
- 討論區管理
- 試卷管理
- 題庫維護
- 分組管理

評量管理

- 成績管理
- 配分設定

【107上】1189計算機概論與程式設計 Intro

Quiz1

- Quiz1\_Q1\_sample.c
- Quiz1\_Q2\_sample.c
- Quiz1.pdf

下載資料夾 編修

點擊下載

下載完記得先解壓縮，再開始編寫程式

# 考試規則

1. 可以翻閱你覺得有幫助的書、講義 (教室會斷網!!)
2. 不得作弊，違者依校規論處
3. 若有格式錯誤的情形，會將該題分數  $\times 0.8$  計算
4. 本次練習都只需繳交 **Header file**  
不得更改 `main_Q1.c` 中任何內容  
繳交時請自行將 Header file 檔名改為 學號－題號  
如：0756704-1.h  
註：不需變更 `ifndef`, `define`, `include` 的檔名
5. 總共只有一**次**繳交機會，請務必確認格式正確後，再舉手找助教繳交。
6. 行動電子產品 (手機、平板電腦等等)請收在包包內，不要放在桌面上或使用它。

# Header file

- Header file contains function declarations and macro definitions to be shared between several source files.
- For example

```
main.c x
1  #include <stdio.h>
2  #include "add.h"
3
4  int main(void)
5  {
6      int number;
7      scanf("%d", &number);
8
9      number = do_something(number);
10
11     printf("%d", number);
12
13     return 0;
14 }
15

Start here x *add.h x
1  #ifndef add_H
2  #define add_H
3
4  int do_something(int n){
5      return n + 1;
6  }
7
8  #endif
9
```

- In Header file (\*.h) , you can add any function or declaration **except** main function

# Q1 - Quadratic equation

## Description

Create a class called Quadratic for performing the Quadratic equation ( $ax^2+bx+c$ ).

Provide the following member functions:

1. *solver()* use the formula  $\frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$  to solve the equation
  - No matter how many solutions, the results are in the form of real and imaginary
2. *print()* print the answers in the form  $m+ni$  , where  $m$  is the real part and  $n$  is the imaginary part

Attributes:

1. The coefficients,  $a$ ,  $b$ , and  $c$
2. The answer,  $m$  and  $n$

All the attributes and functions are using *Double* type

# I/O Format

- Input (All double)

- a b c

- Output (All double)

- m1\_n1

- m2\_n2

- Notice: print  $\frac{-b+\sqrt{b^2-4ac}}{2a}$  first, then  $\frac{-b-\sqrt{b^2-4ac}}{2a}$

```
1 -5 6
3 _ 0
2 _ 0
```

```
1 2 3
-1 2.82843
-1 -2.82843
```

```
1 -2 1
1 _ 0
1 _ 0
```

# Q2 – Circle

## Description

Create a class “Circle” with attributes, coordinates of center, “*center\_x*” and “*center\_y*”, and radius of circle, “*radius*”. Provide member functions that “*cal\_area*” to calculate the area and “*cal\_circumference*” to calculate circumference (圓周長) of the circle. Also, provide set functions to define the attributes of this circle. The rules of set function are defined as follows:

*set\_circle* (float x, float y, float radius) – directly set attributes of circle and print the information of this circle

*set\_circle* (float x, float y, float radius, Circle c2) –

1. if the new circle c1:(x, y, radius) has no contact with c2 or only one contact point, directly set the attributes
2. if the new circle c1 has two contact points with c2, always **shrink** the radius of new circle (c1) to become only one contact point
3. if c1 (x, y) is exactly in the circumference of c2, make new circle be the same with c2



# I/O

## ■ Input format

- *Type: int*
- *Attributes of circle : float*

## ■ Output format

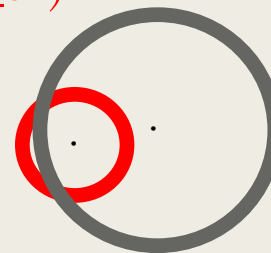
- *All numbers are showed in most 10 digits*

## ■ Notice

- *The radiuses of  $c1$  and  $c2$  are both less than or equal to the distance between two centers  $C1.radius \leq d(\text{center\_c1}, \text{center\_c2})$*

■  $C2.radius \leq d(\text{center\_c1}, \text{center\_c2})$

Don't consider  
these situations



# Example

■ Type 1: only one circle

```
I 1
  3 3 3
O 3_3_3_28.27433395_18.84955597
```

Type

x y radius

x1\_y1\_radius1\_area1\_circumference1

■ Type 2: compare two circles

```
2
0 0 3 5 0 4
0_0_1_3.141592503_6.283185005
5_0_4_50.26548004_25.13274002
```

Type

x1 y1 radius1 x2 y2 radius2

x1\_y1\_radius1\_area1\_circumference1

x2\_y2\_radius2\_area2\_circumference2

Q&A