Course Introduction

Program Design (II)

2022 Spring

Fu-Yin Cherng
Dept. CSIE, National Chung Cheng University

Today’s plan

- Introducing Program Design (II)
- Introducing Program Design (II) Quiz Section

Disclaimer

If you took my course last semester,
you will probably see some of the following slides…

Welcome to the course! ✌

- Instructor 👩‍💻
  - Fu-Yin Cherng (程芙茵) fycherng@cs.ccu.edu.tw

TAs 👩‍🎓

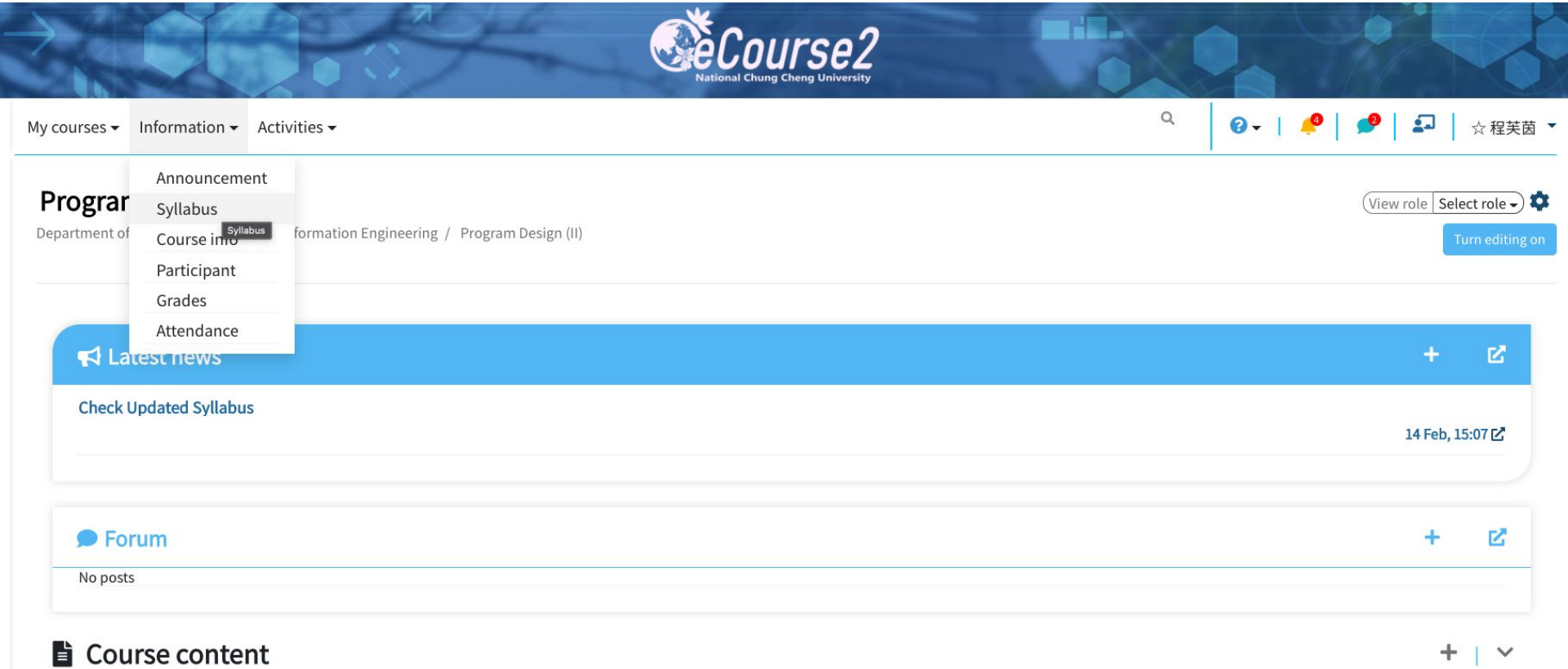
- Program Deisng (II)
  - 顏于婷: graislife8116@gmail.com
- Program Deisng (II) - **Quiz Section**
  - 方星硯　　cocoshen89@csie.io
  - 田雅今　　qooq147852@gmail.com
  - 陳俊翰　　junhanlune@gmail.com

Time and Classroom

- Program Design (II)
  - 10:15-11:30 Tuesday and Thursday
  - Room 001, College of Engineering (I) (工院一館001教室)
- Quiz Section
  - 19:10-21:00 **Tuesday** Room 341, Innovation Building (創新大樓341)
  - 19:10-21:00 **Thursday**. Room 341, Innovation Building (創新大樓341)
  - No quiz section this week!

Course Website

- eCourse2
  - All materials and info. (links to the online lecture) will be announced through this platform
- Let’s have a quick tour on eCourse2


eCourse2
National Chung Cheng University
My courses
Information
Activities
☆程芙茵
Announcement
Syllabus
Syllabus
Course info
Participant
Grades
Attendance
Prograr
Department of
formation Engineering / Program Design (II)
View role
Select role
Turn editing on
Latest news
Check Updated Syllabus
14 Feb, 15:07
Forum
No posts
Course content

CCU Course Syllabus

Program Design (II)程式設計（二）

Course Introduction	This is an introductory course on how to program in C. The scope of this course will extend from the content of Program Design (I). The topic ranges from Strings to File I/O and other miscellaneous functions. This course will also cover some concepts for programmers to develop clean and usable programs and applications. The students who want to enroll in this course need to have basic knowledge of programming. C is a procedural computer programming language and has hugely influenced many programming languages such as C++, Java, and C#, which are all widely used nowadays. Hence, understanding the fundamental concepts of C and knowing how to use C in the program are beneficial to learning most of the new programming languages in the future.
Document of syllabus	110-2 Program Design (II).pdf

Introducing Program Design (II)

4101032_02: Program Design (II)

Spring 2022 (110-2)

Instructors: Fu-Yin Cherng
Department of Computer Science and Information Engineering
National Chung Cheng University

Introduction

This is an introductory course on how to program in C. The scope of this course will extend from the content of Program Design (I). The topic ranges from Strings to File I/O and other miscellaneous functions. This course will also cover some concepts for programmers to develop clean and usable programs and applications. **The students who want to enroll in this course need to have basic knowledge of programming.** C is a procedural computer programming language and has had a huge influence on many programming languages such as C++, Java, and C# which are all widely used nowadays. Hence, understanding the fundamental concepts of C and knowing how to use C in the program are beneficial to learning most of the new programming languages in the future.

Basic information

Instructor.

- Fu-Yin Cherng (程芙茵): fycherng(AT)cs.ccu.edu.tw; https://fuyincherng.github.io/

please contact me by this email address

Teaching Assistants.

- 顏于婷: graislife8116(AT)gmail.com

Lectures. 10:15-11:30 **Tuesday** and **Thursday**. Room 001, College of Engineering (I) (工院一館001教室)

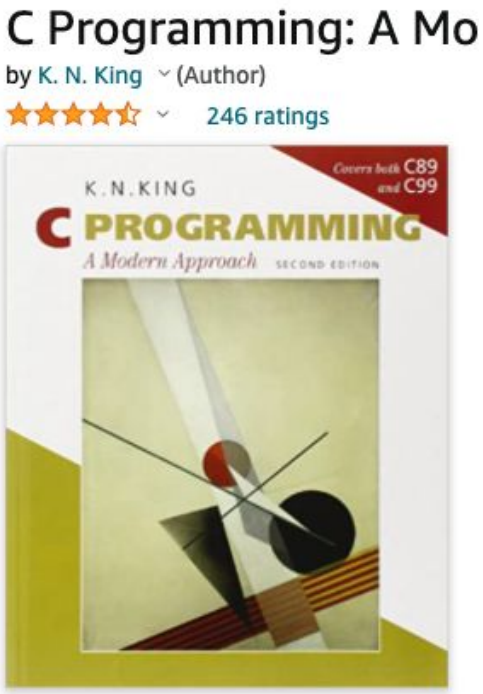
Textbook

C Programming: A Modern Approach by K. N. King, 2nd edition, 2008, W. W. Norton & Company.

- Rich but organized content
- C89 and C99: best-supported version
- Continue from Ch 13 (Strings)

Timeline of language development

Year	C Standard[9]
1972	Birth
1978	K&R C
1989/1990	ANSI C and ISO C
1999	C99
2011	C11
2017	C17
TBD	C2x



"It is thorough but clear, and includes plenty of code examples to learn from." -- Amazon review

Homework (40%)

- Submit homework to DOMjudge
  - However, we will encourage you to upload your homework on Github (one of the must-learns for CS people)
- If you don't know how to use DOMjudge, TAs will explain how to use DOMjudge to submit homework in the first quiz session.

- Homework 1- 4: 40%
  - You will get a half score of your homework if you submit it **one** day after the deadline and you will receive **25% points** if you hand in the homework two days (or more) after the deadline.

Midterm and Final exams (40%)

- Midterm exam (20%) will be paper-and-pencil test (紙筆測驗)
- Final exma (20%) will be programming test (上機考)

Final Project (20%)

- Larger programming project than final project in the last semester
- Group project (team of 2 to 4 students)
  - larger team, larger project
  - Please start to find your teammates as soon as possible
- Final project demo and presentation at the end of this semester
  - presentations in English 😉
- Will announce other detail soon

(Bounus) Class Participation (5%)

- Impression score, for example
  - Discussion in eCourse2 (e.g., asking questions, answering others questions, sharing info)
  - Actively help other students during quiz section and after the class
  - Email to me and TA when you find error in the homework and program exercise
  - Anythings that can demonstrate that you actively participate in this class
- Please make sure your participation is visible to me and TA. Otherwise, we won't know your participation 😂
  - please remind me and TAs if we forgot when calculating the final score

Grading

Breakdown.

- Homework 1- 4: 40%
  - You will get a half score of your homework if you submit it **one** day after the deadline and you will receive **25% points** if you hand in the homework two days (or more) after the deadline.
- Midterm and Final exams: 40%
- Final Project: 20%
- (Bonus) Class Participation: 5%

The TAs will grade everything and regrade them upon request. If you have a regrading request, please contact the TAs directly.

It's actually not that hard to pass this course! 😳

- complete all homework (+40%)
- get at least 25 for midterm exam (+5%) and final exams (+5%)
- get 50 for the final project (+10%)
- actively participate in class by (+4%)
- Total score: 64 out of 100

Tentative Schedule

Week	Date	Note	Lecture	Textbook
1	2/15		Course Introduction	
	2/17		Review of C	
2	2/22		Strings	Ch 13.1 - 13.3
	2/24		Strings	Ch 13.4- 13.7
3	3/1		The preprocessor	Ch 14.1 - 14.3
	3/3		The preprocessor	Ch 14.3 - 14.4
4	3/8	Homework 1	The preprocessor	Ch 14.4 - 14.5
	3/10		Writing Large Programs	Ch 15.1 - 15.2
5	3/15		Writing Large Programs	Ch 15.3 - 15.4
	3/17		Structures, Unions, and Enumerations	Ch 16.1 - 16.2
6	3/22		Structures, Unions, and Enumerations	Ch 16.3
	3/24	Homework 2	Structures, Unions, and Enumerations	Ch 16.4 - 16.5
7	3/29		Advanced Uses of Pointers	Ch 17.1 - 17.2
	3/31	*Holiday*		

Please read the tentative schedule and manage your time accordingly

10	4/19	*remote lesson*	Program Design and User Experience	
	4/21	*remote lesson*	Program Design and User Experience	
11	4/26	Homework 3	Declarations	Ch 18.1 - 18.3
	4/28		Declarations	Ch 18.4 - 18.6
12	5/3		Program Design	Ch 19.1 - 19.2
	5/5		Program Design	Ch 19.3 - 19.4
13	5/10		Program Design	Ch 19.4 - 19.5
	5/12		Low-level Programming	Ch 20.1
14	5/17	Homework 4	Low-level Programming	Ch 20.2 - 20.3
	5/19		The Standard Library	Ch 21
15	5/24		Input/Output	Ch 22.1 - 22.2
	5/26		Input/Output	Ch 22.2 - 22.3
16	5/31		Input/Output	Ch 22.4 - 22.8
	6/2		Error Handling	Ch 24
17	6/7	*remote lesson*	Program Design and User Experience	
	6/9	*remote lesson*	Program Design and User Experience	

Expect to have remote lessons in Week 10 and Week 17.
The link to remoste course videoes will be posted on eCourse2

About the attendance rate (出席率)

- Different people have different ways to learn and you should find what is the best for you.
- Please own your learning and be responsible for your money and time 😎
- So, we don’t count the attendance rate for this course
- However, you need to attend every quiz section

About Plagiarism and Cheating

- Start doing the homework as soon as possible
- Ask for help from your classmates, TAs, instructor, and Google
- It's your responsibility to own your learning! (again, sorry
- Please let us know if the homeworks are too hard for you. You suppose to be able to complete the homework based on what you learn in the lecture.

- Please let us know if you found any case of plagiarism/cheating. We will handle it

Any Questions?

Introducing Program Design (II) - Quiz Section

What do you need to do for the Quiz Section?

First, you will be assigned to a team (1/2)

- Based on the programming test we will announce soon, we will try our best to ensure the students with different levels of programming are distributed evenly in each team.
- You will be on the same team for the entire semester

First, you will be assigned to a team (2/2)

- Basically, a team of eight students.
  - Simulate the conditions in the industry
  - If you go to industrial work in the future, why not practice teamwork as early as possible?
  - Practice using the technical tools to facilitate teamwork (e.g., Trello, Notion)

Second, read the program exercise for this week on eCourse2 and finish it with your teammates before the quiz section next week (1/2)

- We will upload the description of the program exercise on eCourse2 before the quiz section in the current week.
- Ensure you and your teammates read it before the quiz section
- If you have any questions about the program exercise, you can ask TA during the quiz section.

Second, read the program exercise for this week on eCourse2 and finish it before the quiz section next week (2/2)

the rules are a bit different from last semester

- You have a week to finish the program exercise.
- TA will select half of the teammates to demo the program exercise last week during quiz section
- The other half of the teammates will need to review the code
- We will ensure every student will equal opportunity for demo and review

What is code review and why?

- *Code review is a process which a developer's code is examined by a peer. Code reviews help developers discover common bugs faster and reduce the amount of work required to optimize code in the latter stages.*
  - https://google.github.io/eng-practices/review/reviewer/
  - https://www.browserstack.com/guide/code-review-benefits
  - https://blog.alantsai.net/posts/2019/05/code-review-what-is-code-review-and-why-we-want-to-do-it
  - https://buzzorange.com/techorange/2016/08/16/airbnb-code-review
- Although the program exercise in quiz section is simple and small, practicing how to code review as early as possible will have great benefits to students' ability of coding and chances to learn from others.

https://www.browserstack.com/guide/code-review-benefits

Grading *Program Design (II) - Quiz Section*

Breakdown.

- Rate of Attendance: 50%
  - Every student needs to attend every quiz section.
  - Please notify TA of your group if you cannot attend the quiz section this week in advance and show the relevant evidence.
  - **If a student doesn't show up three times without notifying the TA in advance for three times, the student cannot pass the Quiz Section and Program Design (II).**
- Program Exercise and Code Review: 50%
  - TA will select half of the students from a group to demo the program exercise last week. Those half of the students will get the score of this item.
  - The other half of the students will need to review the code and they will get the score of this item.

Basically, you just need to attend every quiz section, finish some exercise and do some code reviews -> you pass! 🤷

Tentative Schedule

Week	Date	Note	Lecture of Program Design (II)	Quiz Section
1	2/15		Course Introduction	**No Quiz Section**
	2/17		Review of C	
2	2/22		Strings	*Setting up Programming Environment of C*
	2/24		Strings	
3	3/1		The preprocessor	Practice of Strings
	3/3		The preprocessor	
4	3/8	Homework 1	The preprocessor	Practice of Preprocessor
	3/10		Writing Large Programs	

Make sure you check the schedule and mark every deadline and event on your calendar (e.g., Google calendar)

About Asking Programming Questions...

- Because there are unlimited tools, it’s impossible for us to be experts in every tool.
- It’s ok to ask programming questions. But we have received many confusing questions in the past.
- When asking programming questions, please offer us sufficient information 😂
  - What do you want to do?
  - Your code, the error message, screenshots, computer configurations
  - https://stackoverflow.com/help/how-to-ask

About Asking Programming Questions...

- Google and Stack Overflow are every programmer's best friend
- The best resources and articles about programming are in English
  - the creator of C uses English, developer of UNIX uses English, the best programmers in the world use English to communicate
  - You write code in English!
- So, that's why we are trying to teach C programming in English
- Don't let the language be an obstacle to your career as a programmer and software developer

學習程式的第一天

學習程式的第十年


Are there any Questions?
Yes?
Use Google