Programming for Engineers
Introduction to the course
Course ID: EE057IU

Instructor: M.Eng Nguyen Minh Thien

Outline

- Instructor's Information

- Roles and Aims of the Course

- Course Catalog Description

- Course Grading Policies

- Course Assignment and Assessments
  - Attendance, Homework/Quizzes, Projects, Exams
  - Collaboration and Academic Integrity Policy

- Reference Books and Handout Materials

- Integrated Development Environment (IDE)

Instructor's Information

📧 **Email:** nmthien@hcmiu.edu.vn

🕐 **Course time**: 13:15-15:45 Wednesday (G1) – Friday (G2)

👥 **Office location:** La2. 109 (RF & Microwave Lab)

📅 **Office hours:**

- Monday (11 AM – 4:30 PM)

Link: About me

Course Grading Policy

Our Grading Policy

Attendance	**10%**
Assignments + Project	**30%**
Midterm Test	**20%**
Final Exam	**40%**
	100%

Course Assignment and Assessments

- **<u>Attendances:</u>** According to university regulations, students are permitted a maximum of 3 absences. Attendance will be recorded at the first 30 minutes.

- **<u>Assignments:</u>** Homework is assigned every lecture. While group discussion is permitted and encouraged, individual work is the basis for grading. Copying and plagiarism are strictly prohibited. No late assignments will be accepted
  - ➤ Well-written code and exceptional solutions will receive extra credit for **Assignments** and **Projects**
  - ➤ If you earn over 100 points on a Homework or Project, any bonus points will be transferred to assignments with lower scores.

Course Assignment and Assessments

- **<u>Project:</u>** There is a large and complex programming project. Students are given ample time to complete it.

- **<u>Test & Exam:</u>** consist of one midterm test and one final exam, both of which are closed book and prohibit the use of the internet. No make-up exam, only allowed in exceptional circumstances.

- **<u>Academic Integrity Policy:</u>** Copying, plagiarism, and cheating are prohibited. Group discussions are encouraged and permitted. While ChatGPT can be used for assistance, students are encouraged to understand the material themselves.

  - ➢Penalties for violation: Warning, lowering grade, failing grades

Plagiarism – Classroom Policy

➤ What is Plagiarism

- Getting code from the Internet without understanding
- Asking someone else to write the code for you
- Copying your friend's code

➤ Classroom policies

- No use of phones/laptop/tablet with loud speakers.
- No crosstalk
- No Food/Drink
- Raise hand to ask questions

Roles and Aims of the Course

➢ **Roles of the course**

Programming is the core of Electrical/Computer Science Engineering.

- **Introduction to C Programming:** Provide foundational knowledge of C.

- **Skill Development:** Equip students with the ability to write, debug, and optimize C code.

- **Practical Application:** Prepare students to apply programming skills to engineering problems and projects.

- This course is considered as the kick-off course in Programming and Computer Science using "enjoyable" C (procedure-oriented) programming language.

Roles and Aims of the Course

➢ **Aims of the course**

- **Master the Basics of C Programming**: Ensure students gain a solid understanding of the fundamental constructs of the C language, including its syntax, structure, and standard libraries.

- **Develop Algorithmic Thinking**: Help students develop the ability to design algorithms and translate them into C code, emphasizing logical flow and problem-solving strategies.

- **Apply C to Engineering Problems**: Enable students to use C programming to address real-world engineering problems, including calculations, data analysis, and system modeling.

- **Encourage Best Practices**: Promote best practices in coding, including writing clean, maintainable code, adhering to coding standards, and using version control systems.

- **Foster Independent Learning**: Encourage students to seek out additional resources, practice programming outside of class, and continuously improve their skills.

Course Catalog

Timeline	Lecture Contents	Material	Notes
Week 2 (08/09-14/09)	Course Overview Introduction C Programming	Lecture Slide_0 – 1A – 1B	Class discussion
Week 3 (15/09-21/09)	Flow Chart/Pseudo code/ Decision making/Data Types	Book, Lecture Slide_2A – 2B	In-class Exercises;
Week 4 (22/09-28/09)	Iteration & Input/Output Operation	Book, Lecture slide_3	In-class Exercises; HW
Week 5 (29/09-05/10)	Function & Recursion	Book, Lecture slide_4	In-class Exercises;
Week 6 (06/10-12/10)	C Arrays	Book, Lecture slide_5	In-class Exercises;
Week 7 (13/10-19/10)	Sorting Algorithms	Book, Lecture slide_6	In-class Exercises; HW
Week 8 (20/10-26/10)	**Midterm Test**	Laptop w/o Internet; Book, Slides	In-class Test

Course Catalog

Timeline	Lecture Contents	Material	Notes
Week 9 - 10 (27/10-09/11)	Pointers & Function Pointers	Book, Lecture Slide_7-8	In-class Exercises;
Week 11 (10/11-16/11)	Characters & Strings	Book, Lecture slide_9	In-class Exercises; HW
Week 12 (17/11-23/11)	Structures/Unions/Enumerates	Book, Lecture slide_10	In-class Exercises;
Week 13 (24/11-30/11	Bitwise-Operation/File Processing/ Dynamic Memory Allocation	Book, Lecture slide_11	In-class Exercises; HW
Week 14 (01/12-07/12)	Linked Lists/Queues/ Stacks/Binary Tree	Book, Lecture slide_12-13	In-class Exercises;
Week 15 (08/12-14/12)	Project presentation		In-class Exercises;
Week 16 (15/12-21/12)	**Revision**		
Week 17&18	**Final Exam**	Laptop w/o Internet; Book, Slides	Follow IU schedule

Reference Books and Handout Materials

- **"*C - How to Program*", 9th edition, Harvey M. Deitel, Paul J. Deitel, Prentice Hall, 2021.**

- "*Programming in C*", 3rd edition, S. G. Kochan, Sams Publishing, 2005.

- "*Programming In C: A Practical Approach*", 2nd edition, Ajay Mittal, Pearson, 2008.

- "*An Introductory to Programming with C*", 2nd edition, Diane Zak, Thomson Publishing, 2001.

- "*Data Structures and Algorithms*", Asian edition, Alfred V. Aho, John E. Hopcroft, Jeffrey D. Ullman.

Integrated Development Environment (IDE)

- **Windows:** VS Code, DevC++, Microsoft Visual Studio (Community), **Eclipse**, **CodeLite**...

- **MacOS:** VS Code, Xcode, **Eclipse**, **CodeLite**...

- **UNIX/Linux:** VS Code, **Eclipse**, **CodeLite**, NetBeans, Atom Code Editor, Clion...