Course Syllabus (Summer 2019)

CET111: Programming Paradigms: The C Programming Language

Course Information

Prerequisite: N.A.

Schedule: Refer to class webpage

Classroom: SR3C

Instructor: Howard Sin

Contact: howard.sin@digipen.edu

Office Hours: Class time or by appointment

Class Webpage: cet111s19-a.sg at http://distance.sg.digipen.edu
The website is accessible via student's DIT login credential.

Description

The module provides the fundamentals of computer programming in general, as well as the introduction to the C programming language, including operators, control flow, functions, and more advanced features. The module covers basic data types, arrays, structures and pointer use and arithmetic. Additionally, it intensely discusses the lexical conventions, syntax notation and semantics of the C programming language.

After completing this module, the students should be able to efficiently and effectively communicate with a computer using an imperative, relatively low-level programming language with a structural and procedural programming paradigm. The students should be able to write various computer programs in the C programming language that implement computer algorithms and solve classical problems, as well as they should be able to compile, link, execute, and debug such a program.

The last part of the module contains a short introduction to the C++ programming language for the C language software developers.

References

- King K. N., "C programming; A Modern Approach", 2nd Edition, W. W. Norton, 2008.
- Kernigan B., Ritchie D., "The C Programming Language", 2nd Edition, Prentice Hall, 1988.
- Harbison S., Steele G. Jr., "C: A Reference Manual", 5th Edition, Prentice Hall, 2002.

Module Topics

Unit 1. Basic Features

- Introduction
- Program Organization
- Compilation Process

- Basic Data Types
- Formatted Input and Output
- Operators, Operands and Expressions
- Statements and Flow Control
- Loops
- Array
- Functions, Stack, Run-time Environment
- Coding Conventions

Unit 2. Advanced Features

- Pointers and Dynamic Data Allocation
- Pointers and Arrays
- Strings of Text
- The Preprocessor
- Writing Large Programs
- Complex Data Types
    - Structures
    - Unions
    - Enumerations
- Declarations
- Low-Level Programming

Unit 3. The Standard C Library

- Overview
- Input and Output
- Numerical and Character Data
- Error Handling
- Standard Functions

Unit 4. Miscellaneous Topics

- Software Design
- Software Development Process and Tools
- Brief Introduction to C++
    - High-level Comparison of C++ and C
    - Overview of Object-Oriented Programming

Grading Policy

Grades will be derived from project, exams and class participation.

The composition of grades is as follows:

- Assignments: 100% total. 20 assignments x 5% each.

Do note that the number of assignments may vary in order to meet the proficiency requirements of the course. In the event of changes, the total grades per category will remain unchanged while each category will update such that grades are evenly distributed across each entry within.

Attendance Policy

Attendance is **mandatory**. Students are expected to **notify the lecturer** if they are unable to attend class for any reason. A 1 week grace will be given before and after the class for the student to submit the notice. It is highly recommended for student to send the notice via email to keep a digital record of the notice.

Each absence case will be reviewed on a case-by-case basis and students may be required to provide document proof to back their reasons given.

Typical valid reasons are those that fall within the domain of medical (with valid medical certificate) and compassion (with document proof).

Any reasons that are deemed to be personal and/or due to negligence will be rejected. Examples are like oversleeping, working on another assignment/project/course, uncertain if class is being held, going on a family holiday… etc.

A **5% penalty** from the course total will be given for each unexcused absence.

Submission Requirement Policy

In order to properly grade works submitted, all source code and assets are required for assessment.

Detailed instructions and requirements will be listed in each assignment submission page.
You are expected to follow all instructions and meet all requirements stated.

Grades will be given based only on what is submitted to the school's course management system (Moodle) and no exceptions will be allowed.

Source, header, data, and README files must start with the following header:

```
/******************************************************************************
filename Example.c
author John Smith
email john.smith@digipen.edu
date created 1 Jan 1988
Brief Description: A simple summary of what the code does
******************************************************************************/
```

Failure to include the mandatory header comment will result in a **20% penalty** for the assignment.

Late Submission Policy

Each assignment provided will be accompanied with a due date and time which will be clearly stated on the assignment submission page. A **100% penalty** will be imposed on any submission deemed late by the course management system (Moodle). This will also be reflected on the assignment brief.

Students may request for extensions should they provide valid reasons with documented proof to justify their case. This will be handled on a case-by-case basis by the lecturer.

Request for extensions after the deadline will not be accepted except in excruciating cases. In those cases, it will be handled on a case-by-case basis by the school.

Classroom Policy

Students are expected to behave professionally at all times with regards to classroom conduct and timely delivery of all assignments. Specific guidelines will accompany each assignment, along with a completion date. Students are expected to retain all works done until after the end of the semester.

To maintain a conducive learning environment during class, it is expected for students to…

- Be quiet during class while the lecturer is talking and to keep a low noise level at all times. This is to ensure that everyone is able to listen to the discussion and to not disturb others while doing class activities.
- Turn off your mobile phones or put them on silent mode. This is to prevent unwanted interruptions due to phones ringing or vibrating.
- Reduce use of mobile phones during class where possible. Also, playing games on any device is strictly prohibited during class time. This is to reduce distractions for everyone around including the student him/herself. Penalties may be imposed if students are caught doing so.
- Keep the classrooms clean. Eating and/or drinking in class is strictly prohibited with the exception of bottled water. Do dispose of all wastes (such as used paper, eraser crumbs, empty bottles etc.) in the garbage bins located outside classrooms.
- Do not mistreat school equipment (such as computers, keyboards, mice, monitors etc.). There will be penalties given for such abuse cases and having broken equipment will cause inconvenience for everyone in school.

Academic Integrity Policy

CET111 assignments are **NOT** group projects. They must represent a student's own individual work. It is reasonable for students for students to consult or discuss general solutions to an assignment. However, it is prohibited for students to collaborate on detailed solutions, to copy code, or to give away code.

Cheating, or academic dishonesty in any form, will not be tolerated in this course. Penalties for cheating may include receiving a zero on an assignment, or a failing grade in the course, or even expulsion from DigiPen. It is permissible to discuss assignments (not solutions) with other students in the class, but the solutions must be recognizably your own. For further details, please consult the DigiPen Academic Integrity Policy.

With the internet as a readily accessible source of information and help, students may feel that plagiarism is ambiguous and thus be unable to determine what it constitutes. Here are some general guidelines to help make the distinction:

- Do **NOT** copy-paste any works online (wholesale or otherwise). Using works that are not yours is plagiarism.
- Do **NOT** ask online communities (such as stack overflow, unity forums etc.) to solve your bugs & code issues by providing your code segments. Asking others to solve your issues is work not done by you and thus is plagiarism.
- You may learn from sources online, understand the workings and concepts, and implementing them again via **your own efforts**. A good habit is to assume that you will be tested on the things you learn online and if you are unable to answer the questions then you should not use said works.
- You may ask online communities general problems and use their insights to **work on your problem**.
- These applies to all sources on any medium (be it the internet, textbooks, friends or social media etc.). It is the content that is important, not the medium they are on.
- The bottom line test is to ask yourself "Did I work on this?" If you did not, then you should not use it. **Learn** from it and **work it out yourself**.