

## **CE1007/CZ1007 Data Structures**



**Hui Siu Cheung**

Tel: 6790-4930

Email: [asschui@ntu.edu.sg](mailto:asschui@ntu.edu.sg)

&



**Tan Kheng Leong**

Tel: 67905193

Email: [khengleong@ntu.edu.sg](mailto:khengleong@ntu.edu.sg)

1

### **CE1007/CZ1007 Data Structures**

1. This course is CE1007/CZ1007 Data Structures.
2. The programming language used in this course for data structures is C. There are two major components for this course: C programming language and Data Structures.
3. For this course, there are two lecturers. They are Dr. Hui Siu Cheung and Mr. Tan Kheng Leong.
4. Dr. Hui will be responsible for the first half of the course covering the fundamental data structures in C Programming, and Mr. Tan will conduct the second half of the course covering advanced data structures such as linked lists, stacks, queues and trees.

## Course Objectives

- **Course Aim:**
  - This core programming course aims to develop your understanding in data structures such as **linked lists**, **stacks**, **queues** and **trees** that are important for building efficient programs, and are essential for other programming and software engineering courses.
- **Intended Learning Outcomes (ILO):**
  - Design and implement simple C programs using basic programming constructs.
  - Design and implement **pointers**, **arrays**, **character strings** and **structures** in C.
  - Design and implement **recursive functions** in C.
  - Design and implement **dynamic memory allocation** in C.
  - Design and implement **linked lists**, **stacks** and **queues** in C.
  - Design and implement **tree structures** in C.

2

### Course Objectives

1. The pre-requisite for this course is CE1003/CZ1003 Computational Thinking or any similar programming experience. You will need to meet this basic requirement in order to learn the concepts effectively in this course,
2. This course aims to develop your understanding in data structures such as **linked lists**, **stacks**, **queues** and **trees** that are important for building efficient programs, and are essential for other programming and software engineering courses.
3. The intended learning outcomes upon completion of the course include the following:
  - Design and implement simple C programs using basic programming constructs.
  - Design and implement pointers, arrays, character strings and structures in C.
  - Design and implement recursive functions in C.
  - Design and implement dynamic memory allocation in C.
  - Design and implement linked lists, stacks and queues in C.
  - Design and implement tree structures in C.

## Learning Activities – Topics for C Programming

- Week 1 – Basic C Programming & Control Flow (Lecture); no lab & tutorial
- Week 2 – Functions (Lecture); Lab 1
- Week 3 – Pointers (Lecture); Lab 2
- Week 4 – Arrays (Lecture); Lab 3
- Week 5 – Character Strings (Lecture); Lab 4
- Week 6 – Structures (Lecture); Lab 5
- Week 7 – Recursive Functions (Lecture); Lab 6
- Recess Week
- Week 8 – Lab Test 1 (**12 Oct 2019, Sat**)
- Week 8-13 – C Data Structures (by Tan Kheng Leong)

3

### Learning Activities

1. The topics to be covered in the first half of the course include basic C programming, control flow, functions and pointers, arrays, character strings, structures and recursive functions.
2. Please note that Lab Test 1 will be held on week 8, 12 Oct 2019, Saturday. Please mark this date in your calendar and prepare for the lab test.
3. The second half of lectures will start from week 8 and will be conducted by the other lecturer.

## Course Structure

- Course Mode –TEL video lectures + scheduled Lectures
  - Please read lecture slides, lecture slides with notes, and watch TEL video lectures
- Scheduled Lectures
  - 1 lecture hour/week
  - Start from **Week 1**
- Lab –Tutorial
  - 1 lab-tutorial/week, 2 hours/week [according to your lab tutorial groups]
  - Start from **Week 2**
  - Lab Tutors are available for consultation in the lab sessions

4

### Course Structure

1. The course mode is in the form of TEL video lectures and scheduled lectures. Please read the lecture slides and lecture notes, and watch TEL video lectures for learning the course contents.
2. For scheduled lectures, there will be a total of 13 lectures, and 1 lecture hour per week.
3. For lab-tutorial sessions, there will be one lab hour and one tutorial hour (i.e., 2 consecutive hours) per session for each week. Please note that lab-tutorial sessions will start from **week 2**. Please also note that it is important to attend the lab-tutorial sessions as this course is conducted as a practical programming course.

## Course Materials – NTULearn

- Course materials are available in NTULearn. For the C Programming part, the key folders in the course site include:
  - **TEL Lecture Video**: This folder contains TEL lecture videos which are constructed for e-learning purpose.
  - **TEL Lecture Video PPT Slides and Notes**: This folder contains the slides and notes for the TEL lecture videos.
  - **Lectures**: This folder contains lecture slides (and notes) used in the scheduled lectures.
  - **Labs and Tutorials**: This folder contains lab and tutorial questions.
  - **Assignments**: This folder contains questions for assignments.
  - **Practice Questions**: This folder contains practice questions for topical concepts.
  - **Lab Test 1**: This folder contains relevant information for lab test 1.
  - **Suggested Code for Lab-Tutorial Questions**: Suggested code for lab-tutorial questions will be posted in this folder after each lab-tutorial session.

5

### Course Materials – NTULearn

1. Course materials are available in NTULearn. You may download them from NTULearn before the lectures and lab-tutorial sessions. The key folders in the course site in NTULearn include:
  - 1) TEL Lecture Videos: This folder contains TEL lecture videos which are constructed for e-learning purpose.
  - 2) TEL Lecture Video PPT Slides and Notes: This folder contains the slides and notes for the TEL lecture videos.
  - 3) Lectures: This folder contains lecture slides (and notes) used in the scheduled lectures.
  - 4) Labs and Tutorials: This folder contains lab and tutorial questions.
  - 5) Assignments: This folder contains questions for assignments.
  - 6) Practice Questions: This folder contains practice questions for topical concepts.
  - 7) Lab Test 1: This folder contains relevant information for lab test 1.
  - 8) Suggested Code for Lab-Tutorial Questions: Suggested code for lab-tutorial questions will be posted in this folder after each lab-tutorial session.

## Course Assessment

- No final written examination
- 100% Continuous Assessment
- Lab Attendance (5%)
  - $\geq 90\%$  lab-tutorial sessions (5%);  $\geq 80\%$  (4%);  $\geq 70\%$  (3%);  $\geq 60\%$  (2%);  $\geq 50\%$  (1%);  $< 50\%$  (0%).
- Assignments (25%)
  - Assignment questions will be posted for each week, and you are required to submit your code for grading [start from **Week 3 - Arrays**]
- Lab Tests (70%)
  - Lab Test 1 (30%) - **12 Oct 2019 (Sat)**
  - Lab Test 2 (40%) - **to be announced later by the other lecturer**

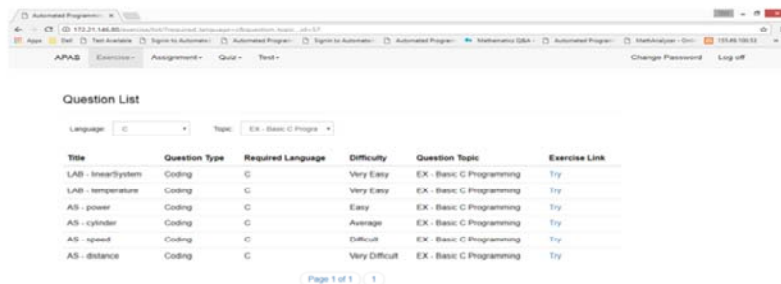
6

### Course Assessment

1. There is no final written examination in this course.
2. We use 100% continuous assessment for course assessment.
3. There are three components for course assessment:
  - 5% for Lab attendance
    - If attending  $\geq 90\%$  lab-tutorial sessions, you will get 5% score; or
    - If attending  $\geq 80\%$  lab-tutorial sessions, you will get 4% score; or
    - If attending  $\geq 70\%$  lab-tutorial sessions, you will get 3% score; or
    - If attending  $\geq 60\%$  lab-tutorial sessions, you will get 2% score; or
    - If attending  $\geq 50\%$  lab-tutorial sessions, you will get 1% score; or
    - If attending  $< 50\%$  lab-tutorial sessions, then 0% score.
  - 25% for Assignments
    - Assignment questions will be posted for each week, and you are required to submit your code for grading [start from **Week 3 - Arrays**]
  - 70% for Lab Tests
    - There will be two lab tests – Lab Test 1 (30%) and Lab Test 2 (40%).
    - Lab test 1 is scheduled to be held on 12 Oct 2019, Saturday.
    - Lab test 2 will be announced later by the other lecturer.

## Automatic Programming Assessment System (APAS)

- An online Automatic Programming Assessment System (APAS) has been developed for this course.
- Assignment/Lab Test will be submitted to APAS, and marked automatically.
- User Id and Password for accessing the system will be given in Week 3.



7

### Automatic Programming Assessment System

1. An online Automatic Programming Assessment System or APAS has been developed and used for this course.
2. Programming assignments and lab test code will be submitted through the APAS system for automated grading. That is, the APAS system will check your submitted code automatically. This is done through the checking on predefined test cases.
3. The user id and password for accessing the APAS system will be given in week 3.

## Integrated Development Environment (IDE)

- Integrated development environment **Code::Blocks** will be used in this course.
- Code::Blocks is open source – you may download it on your PC: Windows or Mac.
- During Lab Test, **only** Code::Blocks will be available for the development of your code.
- **Caution:** Some students did not do well in lab tests due to their lack of practice using Code::Blocks. My advice is that please try to use Code::Blocks in all your program development for your assignments and labs. So that you can use it comfortably for your code development.

8

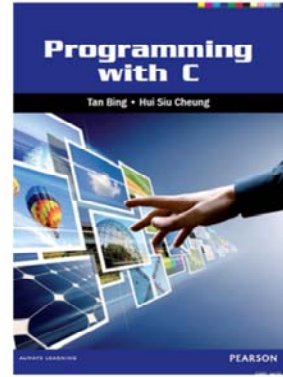
### Integrated Development Environment

1. In this course, we will use Code::Blocks as the integrated development environment for program development.
2. Code::Blocks is open source. You may download Code::Blocks on your personal computers for Windows or Mac .
3. Please note that **only** Code::Blocks will be available for the development of code in our lab environment including lab test.
4. From our past experience, some students did not do well in lab tests due to their lack of practice using Code::Blocks. My advice is that please try to use Code::Blocks in all your program development for assignments and labs. So that you can use it comfortably for code development in lab tests.



## References for C Programming

- For C programming part, we will use the following reference book: “**Programming with C**”, B. Tan and S.C. Hui, Prentice Hall, 2011.
- For C data structures part, the lecturer will let you know the reference book accordingly.
- However, you may use **Online Resources** from the Internet (via the search engine Google)
- Also, the **stackoverflow Q&A website** which provides many useful tips for building programs: <http://stackoverflow.com/>



9

### References and Online Resources for C Programming

1. The reference book “Programming with C” is used for the first part of the course. You may purchase a copy of the book from NTU bookstore or find a copy of the book in NTU library.
2. For C data structures part, the lecturer will let you know the reference book accordingly.
3. There are also many online resources available on C programming and data structures. You may Google for the related C resources. They are equally good for this course.
4. In addition, if you have any questions on understanding the concepts, you may refer to the stackoverflow (<http://stackoverflow.com/>) website to search for related code or possible answers.