School of Computer Science and Electronics Engineering
Data Structures – ITP20001/ECE20010

Contents

General Information

Class Meeting Information

Section	Credit	Day and Hours	Lecture Room	Language
03	3	Mon, Thu 9:30 – 11:15	NTH 311	Lectures in Korean and PPT/Exams&/Quizzes in English

Instructor

Name	Youngsup Kim
Office Hours	김영길 Grace School #201 with an appointment.
Contacts	idebtor@gmail.com (Anytime), 010-4939-2819 (Emergency only)
Piazza	Use for the public open questions and comments, or an option "Post to Instructors".
TA & Tutor	김성빈 010-5836-3884 22100113@handong.ac.kr, 김성민 010-6675-6555 glass@handong.ac.kr Coding Hour at Coding Space: **Thu/Fri. 7:00 ~ 8:30 PM**

Course Description and Goals

Catalog Description – 3 Credit Hours

This course covers some of the general-purpose data structures and includes some basics of algorithms. It aims helping students understand the reasons for choosing structures or algorithms for software development. Topics covered include managing abstract data types, time complexity, linked list, stack, queue, tree, heap, sorting, hash, and graphs. Students learn a systematic approach to organizing, writing, and debugging medium-sized programs through a useful set of algorithmic data structures. They learn to develop useful data structures for organizing, representing data to solve real problems and practicing C/C++ coding skills.

Prerequisites

Students are required to be familiar with C programming language, but not C++.

Objectives

1. Learn the basic C/C++ programming skills such as pointers, array, dynamic memory allocation, recursion, overloading and a bit of Object-Oriented programming as well.
2. Understand the concepts of algorithm, abstraction, and time complexity
3. Program data structures such as stack, queue, linked list, tree, heap, sorting and graph
4. Get familiar with the command-line based programming environment (gnu g++) as well as IDE(Interactive Development Environment) such as MS Visual Studio.

Program Outcomes

PO1 - Scientific Base: an ability to apply the knowledge and information of math, science and engineering

PO2 - an ability to design and conduct experiments, as well as to analyze and interpret data

My Own Objectives

1. Study hard to share.

2. Give a fish, and you feed him for a day; teach a man to fish and you feed him for a lifetime.

Texts, Materials, and Resources

Required Textbook: None

Video Lectures

There are many lectures on data structures subjects available on YouTube.

- Beginning C Programming by Bluefever
- **C++ Programming in One Video** by **Derek Banas** – One-hour C++ introduction video.
- C++ Tutorial – A new tutorial series by Derek Banas on YouTube.
  I recommend lessons named Tutorial, Tutorial 2 ~ 8 and 10 (excluding Tutorial 9)

Joining Piazza (LMS) is required.

To join Piazza, go the www.piazza.com and follow the instruction to register. If you cannot get in, email me, and then I will invite you.

- School: Handong Global University
- Course: ECE20010 Data Structures and C++ for C Coders

Most of our communication between us will go through this site. You should post your questions here. Then your peers, TA or I will answer them. The average response time is about 15 minutes more or less, if we all work together and help each other.

IDE(Integrated Development Environment)

They are the worst tools if you want to be a professional programmer because they hide what is going on from you, and your job is to know what is going on. An IDE, or "Integrated Development Environment" may turn you stupid. They are useful if you are trying to get something done quickly, but not for learning to code at the beginning, they are pointless.

- Use GNU C Compiler (GCC g++), Code, Command lines, Makefile
- Code or MS Visual Studio Community Edition 2022 on Windows and XCode on Mac

GitHub – the place we will go every day during this semester.

https://github.com/idebtor/nowic

- Select "Watch" and "Star" buttons at the top of the github page.
- Select and read "README" first.
- **Select and read "GettingStarted"** and follow the instructions to get started this course.
  You may see the following topics and more:
  - Install MSYS2 **first** and install mingw-w64 to use GNU Compiler Collection(GCC)
  - Get an user's account on github.com
  - Install "Git" and "GitHub Desktop"

Exams, PSets and Grading

Quizzes and Exams

One midterm and one final exam, and pop quizzes without a prior notice. You may expect to have about a quiz, a project or a kind of test whenever every major topic is completed.

Class Participation, Teamwork, and Q/A's on Piazza

Proactive class interaction and teamwork are expected. You are encouraged to post your questions such as homework questions, debugging, errors, anything that other students may also be concerned as well. You may post some recommended resources you have found and share with your colleagues such as websites, tips, video lectures. Also, you are encouraged to help your peers by answering questions on Piazza.

Psets or Problem sets - programming assignments

This course expects many hours of programming and you'll work on your own. Programming assignments will be given almost every week. Upload your file(s) at least **by one hour before the midnight on its due**. Don't ask me one-minute or one-hour excuse. **Follow TA's instructions if any**.

Grading

Grades will be assigned based on the following weights:

Psets(Problem sets), Homework	45
About 10 wake-up pop-quizzes and Labs	5~8
Midterm, Final	23, 23
-0.5 per tardiness, -1.0 per absence	-5
Total	100

Letter grades will be assigned using the following scale:

Grade		+
A	90.0	95.0
B	80.0	85.0
C	70.0	75.0
D	60.0	65.0
F	Below 60.0	

- **Study hard to give:**
  We may have labs and pop-quizzes during the classes, especially, in the beginning of the semester. When we meet in class in person, two students team up loosely, study together, and take quizzes and do the labs and help each other.
- If you do not agree with my grading policy, you should let me know at the first week. **At the discretion of the instructor, grades may be "curved."**

Policies and Advice

Classroom Seat - n/a (not applicable this semester)

Within a week or two after the term begins, your seat will be fixed for the semester. We may try another seat shuffling, if majority of students wish, for the second half of the semester.

Late Work

There will be 25% late penalty for the first 24 hours. No credit after 24 hours of the due date.

Absences

Attendance matters. Two tardy and two absent are excused without penalty. I would not consider your oversleeping, hangover, birthday, cold, or body ache considered as an excuse.

Collaboration and Cheating

All incidents of cheating will be reported to the Office of Student Affairs, who will maintain records of your academic misconduct.

1. Never have a copy of someone else's program in your possession either electronically or on paper and never give your program to someone else.
2. Discussing an assignment without sharing any code is generally acceptable. Helping someone to interpret a compiler error message is an example of permissible collaboration. However, if you get a significant idea from someone or internet sources, acknowledge them in your assignment.
3. No cheatings whatsoever in exams and quizzes.
4. In group projects (if any), you share code freely within your team, but not between teams. Each individual in a team is responsible for the entire project.
5. Cheating on homework or project will lower your letter grade by one at the first time. Cheating on an exam, project or cheating twice in any way, will earn you an F in the course. I reserve the right to assign an F in the course to anyone who cheats even once, though I might not exercise it.
6. Never post a complete program on Piazza for help or question, but a line of code that causes an error. In that case, you do not forget to post the entire error message along with a line of code.
7. You must include the following line at the top of your source file with your name signed. On my honor, I pledge that I have neither received nor provided improper assistance in the completion of this programming assignment. Signed: _____

Advice

In learning programming, a must is **to practice (which means both coding and debugging)**. As you read the lecture notes, try out the examples. Moreover, if you are unsure how some new construct works, write a small sample program and see! **If you approach the course by saying, "I will have fun learning to think in new ways" then you will do well. If you instead say, "I will go through this course and manage to get a pass grade." then you will get frustrated.**

Reservation of Rights

I reserve the right to change this syllabus, including without limitation, these policies, without prior notice.

Weekly Course Schedule

We are going to build this table as we progress this course.

Wk	Topics and Contents	Quiz, Homework, Handouts
1	Chapter 1: Introduction	pset – GNU C/C++, Git/Github/GitHub Desktop, Piazza
2	Chapter 2: Basic Concepts	pset – C/C++, hello.cpp
3	Chapter 3: Arrays and Structures	pset – recursion, function pointers
4	Chapter 3: Arrays and Structures	pset – recursion, function pointers
5	Chapter 4: C++, Sorting, Library	pset – binary search, static library
6	Chapter 5: Stacks and Queues	pset – recurrence
7	Chapter 6: Linked List	pset – profiling, asymptotic notation
8	Midterm exam	pset – stack, queue
9	Chapter 6: Doubly Linked List	pset – infix
10	Chapter 7: Trees	pset – singly linked list
11	Chapter 7: Binary Search Tree	pset – doubly linked list
12	Chapter 7: AVL Trees	pset – tree
13	Chapter 8: Heap, Heap sort	pset – heap, heap sort
14	Chapter 8: Priory Queues	pset – priority queue
15	Chapter 9: Hashing	pset – Hashing
16	Final Exam	
	(Optional) Chapter 10: Graphs	pset - graph

Things to do during the first week:

1. **Read and follow instructions in** https://github.com/idebtor/nowic/01GettingStarted
2. Join Piazza. (www.piazza.com)
   1 주차 월요일 강의 출석체크는 당일까지 Piazza 등록한 것으로 진행합니다. (결석은 -0.5 점)
   Using your ~@handong.edu or ~@handong.ac.kr account, you may enroll in Piazza by yourself. I can do it for you if your email address provided.
3. Install **MSYS2 first. Then install** mingw-w64 to use GNU Compiler Collection.
   After its installation with default settings, you would see one of many folders shown below:
   C:\msys64\mingw64\bin
4. Install "Git" and "GitHub Desktop" and clone github/idebtor/nowic repository.
5. Install VS Code and get familiar with it.
6. Using VS Code, write hello.cpp that prints "Hello World!" on the console (or terminal). Compile it with gcc (gnu compiler collection). You may use the following commands.

   g++ hello.cpp –o hello (to compile and link = to build the executable)
   hello (to execute)

7. We are going to use Piazza folder for your homework submissions.
8. For further study of c programming basics, watch the following lectures on YouTube.
   (1) Beginning C Programming by Bluefever
   (2) C++ Programming in One Video and/or C++ Tutorial **by Derek Banas**.
9. Finish lab1~3 and Start PSet1.

Written by Youngsup Kim (idebtor@gmail.com)