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

Contents

General Information

Class Meeting Information

Section	Days	Time	Credit Hours	Lecture Room	
02	Mon, Thu	09:30 – 11:15	3	NTH 311	Lecture in Korean, Material/Exam/Quiz in English
03	Mon, Thu	16:00 – 17:45	3	NTH 311	

Instructor

Name	Youngsup Kim
Office & Office Hours	NTH 407, Mon, Thu 11:15~12:15, 17:45 ~ 18:15 or with an appointment. Ask your questions on the discussion group available from Piazza.com
Email, Phones	idebtor@gmail.com 260-1171, (010-4939-2819; Texting is acceptable.)
TA	김석진 010-9268-8169, rkathrwks@gmail.com, 최진영 010-4305-2440, 21700750@handong.edu

Course Description and Goals

Catalog Description

This course covers some of the general-purpose data structures and includes some basics of algorithms. It is aimed at helping you 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 and representing data to solve real problems. They are also provided with many chances to practice C/C++ programming skill.

Prerequisites

Students taking this course 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. Be able to 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

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 – Yet to be determined

Reference, not an official textbook: Fundamentals of Data Structures in C++, 2nd Edition, by Horowitz, …

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** - This is a quick one-hour C++ introduction video.
- **C++ Tutorial** – A new tutorial series by **Derek Banas** on YouTube.
  I recommend that you go through lessons named **Tutorial, Tutorial 2 ~ 8 and 10 (excluding Tutorial 9).**

Joining Piazza Discussion Group is required.

There are two ways to join Piazza, go the www.piazza.com.

1. If you have an email address that ends with ~.hgu.edu or ~.handong.edu domain and use it everyday, go the www.piazza.com and follow the instructions in the website.
   To join Piazza, you may need the following information and
   School: Handong Global University
   Course: Data Structures and C++ C coders
2. On your request with your email address, I may register it for you.

We'll be conducting all class-related discussion here this term. The quicker you begin asking questions on Piazza (rather than via emails), the quicker you'll benefit from the collective knowledge of your classmates and instructors. We encourage you to ask questions when you're struggling to understand a concept—you can even do so anonymously.

Install GitHub Desktop

After installation of GitHub-Desktop be a member if already not. From GitHub Desktop, clone the following `nowic` repository into your local computer.

- Clone GitHub/idebtor/nowic repository.
- How to clone a repository from GitHub: Refer to [this site]

Installing Atom

Atom is a text editor that most professional programmers love nowadays.
Some essential packages recommended installing at Atom

- Autosave, File-icons, Mini-maps,
- Platformio-ide-terminal
  - Use + sign at the bottom left corner of the editor

How about IDE(Integrated Development Environment)?

They are the worst tools if you want to be a good programmer because they hide what's going on from you, and your job is to know what's going on. An IDE, or "Integrated Development Environment" will turn you stupid. They are useful if you're trying to get something done and the platform is designed around a particular IDE, but for learning to coding at the beginning they are pointless. **Do not use an IDE for one month or more.**

- We are going to use GNU C Compiler (g++) in MinGW/MSYS available on Windows for several weeks.
- Text editor – Atom

Once you understand the basics of programming using g++, we will use an IDE. Recently I reviewed Dev-C and Eclipse, they have too many things to be desired. I decided to stick to **MS Visual Studio Community Edition** or newer for the time being, unfortunately.

Exams, Projects 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 chapter 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.

Projects or Programming Assignments

Technically, this course consists of two lectures per week and expecting many hours of programming --which simply means you'll work on your own. Programming assignments will be given almost every week. You should upload your file(s) at least **by one hour before the midnight on its due date**. You should follow the following guidelines in packaging your programming assignment.

- **First, follow the instruction if any**.

Grading

Grades will be assigned based on the following weights:

Homework Assignments, Projects	~40
About 10 wake-up pop-quizzes	~10
Midterm, Final	25, 25
-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	

- **Pop-quizzes:** In the beginning of the semester, two students team up, study together, and help each other. **Pop-Quizzes will be graded by a team, not by individuals.** If a student drops out, he/she may pair up if available and agreeable.
- If you don't agree with my grading policy, you should let me know at the first week or day of registration such that I may suggest you an alternative or you should seriously think about options of changing the class or dropping the course. **At the discretion of the instructor, grades may be "curved."**

Policies and Advice

Classroom Seat

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

Late work will not be accepted.

Absences

Attendance will be checked from time to time. There will be a penalty for a missing class or late class attendance. **Two tardy and two absences including job interview and sickness are allowed without penalty.** Oversleeping, hangover, birthday, cold, or body ache would not be 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 which causes an error. In that case, you don't forget posting the entire error message along with a line of code.

7. You must include the following line at the top of your every 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 and to practice. As you read through the lecture notes, try out the examples. And if you're unsure how some new construct works, write a small sample program and see! Find tutorial websites that guide you through the features of subjects.
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	Reading	Quiz, Homework, Handouts
1	Chapter 1: Basic Concepts	Why C++? Debugging	Handout: Syllabus, JumpStart Pset01 – HelloWho
2	Chapter 1: Basic Concepts Chapter 2: Arrays and Structures	SelectionSort Recursion	
3	Chapter 3: Arrays and Structures	Performance Analysis	
4	Chapter 4: Stacks and Queues		
5	Chapter4: Linked Lists		
6	Chapter 4: Linked List		
7	Chapter 5: Trees		
8	Midterm exam		Pset07 – Linked List
9	Chapter 5: Trees		
10	Chapter 5: Trees		Pset08 – Binary Search Tree
11	Chapter 9: Priory Queues		Pset09 – Heap Sort, Priority Queue
12	Chapter 6: Graphs		
13	Chapter 6: Graphs		Pset10 - Graph
14	Chapter 7: Sorting		
15	Chapter 8: Hashing		
16	**Final Exam**	Youngsup Kim	

Things to do during the first week: **Read** https://github.com/idebtor/nowic/GettingStarted

1. Join Piazza. (www.piazza.com)
   Using your ~@handong.edu or ~@hgu.edu email address, you may enroll in Piazza by yourself.
   I can do it for you if your email address provided.
2. Install Desktop-GitHub and clone GitHub/idebtor/nowic repository.
3. Install **Atom** text editor.
4. Using a text editor, write hello.cpp that prints "Hello World!" on the console window(or terminal).
   Compile it with gcc (gnu compiler collection) c++ compiler. You may use the following commands.
   g++ hello.cpp -o hello (to compile)
   hello (to execute)
5. We are going to use Piazza folder for your homework submissions.
6. 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.**
7. Bring your notebook computer during class for a couple of weeks.

Written by Youngsup Kim (idebtor@gmail.com)