U10P32002: DATA STRUCTURE AND ALGORITHM EXPERIMENTS
Spring 2019

Instructors:
Xue Wang email: xwang@nwpu.eud.cn
Time/place:

1. Assignment

There will (likely) be eight programming assignments. They will generally be due one week after they are given out. These assignments are governed by the collaboration policy, given below.

	RELEASED	DUE DATE
PROGRAMMING ASSIGNMENT 1	March 18	March 25

All assignments will be due at 11:59pm on the due date. The reports of all assignments should be turned in electronically, as PDF files. About the report format please see next section below.

2. Data Structure and Algorithm Programming Assignment Report Guidelines

A programming assignment report is a self-contained description of a programming assignment and your solution. It is intended to both show that you learned what the program was designed to teach you, and to instill in you a discipline of rigorous program development and documentation. The program code itself is only a part of such documentation. In addition, you must describe the other important aspects of the program development process, including a description of the problem, design choices, test results, and bugs.

The report must be grammatically correct and use complete English sentences.

Programming Assignment Report Format

For each programming assignment, the corresponding report will include the following sections (when applicable), in the order given. Note that the program code itself is an *attachment* to the main report.

- **Vital information** – This section provides information necessary for properly crediting you for the assignment. Include a section as it appears below, with appropriate items substituted for the items in brackets.
  (COURSE_NUMBER), (QUARTER), (YEAR)
  (YOUR_FIRST_AND_LAST_NAME), (STUDENT_ID)
  Programming Assignment (NUMBER)
  Due (DATE)

- **Problem Statement** – Include a brief summary of the problem and/or your tasks to be completed in this assignment.

- **Structure Chart** – Show your design rationale by creating a structure chart that indicates your top-down, stepwise refinement of the problem solution. You may create the structure chart by hand if you wish (use pen) but there are a number of automated graphical tools available, e.g., Microsoft Office Word, PowerPoint, Visio.

- **Implementation** – Each of the boxes in the structure chart will ordinarily represent a subtask to be performed by a subprogram. In the "Implementation" part of your report you will describe each subprogram in English. *Do not simply include your subprogram code in this section!* The code is handed in as an *attachment* to the main body of the report. Instead, include the following information for each subprogram:
  - Subprogram name
  - Names and types of any parameters
  - Input: Description of the data received
  - Output: Description of the information returned
  - Algorithm: How the subprogram accomplishes the task using *pseudocode*

- **Test Description and Results** – This section describes the tests you ran to determine if your program accurately solves the stated problem. Describe your tests, summarize your results, and argue that they cover all types of program behavior. Often the program handout will require your program to handle certain tests, but you may wish to add others if they bolster your claim that the program is correct. *Do not simply include a script of your program running!* Again, that is handed in as an *attachment* to the main body of the report. You must *write* this test description. After all, you are handing in a report, not just a script. Screenshots of the testing process showing all results can be displayed in this section too. Marking screenshots by annotating, highlighting, or underlining so the reader can easily follow the results. This section should also include a description of any program bugs that is, tests which have incorrect results. The grader will tend to be more lenient if you openly describe bugs and suggest why they may be occurring, than if you simply report them without any analysis, or worse, try to hide them. The ideal test plan and results should show that, despite your best efforts, your tests failed to uncover any bugs.

- **Epilogue** – What were the trouble spots in completing this assignment? What parts caused you the most grief? What did you like about the assignment? What did you learn from it?

- **Attachments** – Include well documented program code to your report, by submitting required source code files. Different assignments may have different submission requirements. At the very least, the program must satisfy the following:
  - A comment should appear just after the program heading which includes the *Vital Information* and *Problem Statement* from the report.

- An informative comment should accompany each global and local variable appearing in the program.
- A comment should appear after each subprogram heading specifying the *Task*, *Input* and *Output* of the subprogram.

- **Acknowledgement** – List the names of other students you may discuss with in the planning or design of solutions to programming assignments. Specify in what aspect collaboration with others works or inspires you most.

3. Collaboration Policy

I would like to emphasize the rules on working with others on these programming assignments. For all these programming assignments, limited collaboration in planning and thinking through solutions to problems is allowed, but no collaboration is allowed in writing up the report or coding solutions. You are allowed to work with other students currently taking U10P32002 in discussing, brainstorming, and verbally walking through solutions to problems. There is no size limit regarding the number of such students you can discuss problems with. But when you are through talking, you must code your solutions and write up your report independently and may not check them against each other. There may be no passing of programming assignment reports between collaborators.

If you collaborate with other students in the planning and design of solutions to programming problems, then you should give their names on the last page of your report as *Acknowledgment* section.

Under no circumstances may you use code obtained from an online or other 3rd party source. All code you submit must be code that you yourself write. The purpose of programming assignments is, aside from helping you learn, testing your ability to come up with and implement correct and efficient algorithms for various problems. Engaging in activities in this paragraph to raise your score while circumventing the spirit of this purpose may be viewed as cheating and result in disciplinary action.

Violation of these rules may be grounds for giving no credit for a programming assignment and also for serious disciplinary action. Severe punishments will apply, so please do not violate these rules. If you have any questions about these rules, please ask an instructor.