COP 2535: Data Structures, Term Project

Step 02, Iteration 1, Design

1 Introduction

The COP 2535 Term Project will give you the opportunity to build an application using one of the data structures we will learn about. You may in fact use several. This project is not intended to be a heavy-weight project, and I don't expect you to make an extraordinary on it. However, this could be part of your portfolio, so you could decide to make more of an effort if you choose to do so.

This project follows an iterative, incremental development model. It is *iterative* in that you will repeat the same series of steps four times through the term. It is *incremental* in that you will add to the functionality of your project during each of the four iterations. Each iteration will consist of the same four phases.

Requirements Analysis In this phase, you will determine the functional requirements to be implemented in this iteration. *Functional requirements* refers to what the software will actually do. This is probably the most important phase, because you cannot build anything unless you know what it is that you are building. Your deliverable will typically be one or more Use Cases, or a Functional Requirements Specification.

Design In this phase, you will design the software to be implemented. In some respects, this is the hardest phase. Thhis is where you will make decisions as to *how* to implement the requirements. Your deliverable will typically be in the form of UML diagrams. We will mostly be using Activty Diagrams, which are similar to flow charts.

Implementation In this phase, you will implement your design in code. The deliverable is your source code.

Testing In this phase, you will test your implementation against the requirements. That is, you will answer the question, "Does the code do what the requirements expect?" Your deliverable will typically be a text document containing the output of the code when run.

2 Instructions

Choose a project. We will discuss some examples in class. Your project should meet the following requirements:

- It should be very simple.
- It should have a text user interface.
- It should focus on maintaining, creating, modifying, deleting, and retrieving data.

Here are some projects you might want to think about:

- A check book balancer
- A music/book/gun collection
- A daily/weekly/monthly scheduler
- A family tree
- A job application tracker

3 Deliverable

Common UML design documents consist of Class Diagrams, System Sequence Diagrams, State Diagrams, and Activity Diagrams. To keep things simple, we will only use activity diagrams. Later on we will explore some other kinds of UML diagrams.

An *Activity Diagram* is similar to a flow chart. (If you do not know what a flow chart is, that's okay — we will not be discussing flow charts.) An activity diagram shows the flow of a program or a component from its beginning to its end. It helps to visualize exactly how a component or a block of code is written.

Your deliverable is a activity diagram. That is, it's an image file, preferable a PDF. Take your use case that you developed in Step 1, and create an activity diagram showing how the use case will be coded in a program.

- `https://modernanalyst.com/Resources/Articles/tabid/115/ID/1830/An-Introduction-to-Activity-Diagram aspx`
- `https://www.lucidchart.com/pages/uml-activity-diagram`
- `https://support.microsoft.com/en-us/office/create-a-uml-activity-diagram-19745dae-2872-4455-a906-1`