Lecture Examples: Code Examples from Class Video Lectures

COSC 2336: Data Structures and Algorithms

Summer 2021

About this Repository

Lecture video code examples for COSC 2336 Data Structures and algorithms class Texas A&M University - Commerce.

This repository should contain all of the example code used in our COSC 2336 Data Structures and Algorithms lecture videos. Examples are numbered by unit number `U01`, `U02`, etc. though some semesters units are rearranged or combined (especially in summer) so you may have slightly different unit numbers than for your class management system. But hopefully the example titles will provide enough information to find and run the code examples you are interested in from the lecture videos.

Built With

This repository assumes you have the same development tools and environment we use for class assignments.

- Gnu C/C++ Compiler Tools
- Gnu Build tools, like Make
- Git revision control tools

Getting Started

This repository uses a standard

```
$ ./configure
$ make
```

to configure and make all of the code examples. There are no tests for the example lecture code, each executable is itself a test/demonstration of concepts from the corresponding lecture video. You only need to run the configuration the first time you want to use the code repository, as usual for our assignments and other activities.

This repository has a `.vscode` setup, so if you are using Visual Studio Code development IDE, as is common for this class, you should have the shortcuts defined from your ide to run the actions:

```
ctrl-shift-1  make clean
ctrl-shift-2  make all
```

If you want to (re)compile a single example, refer to it by name as a make target and run it from the command line as follows:

```
$ make u01-1
$ ./bin/u01-1
```

though of course the `make all` target will only recompile out of date examples, so you can perform make all if you are modifying and exploring one of the code examples to understand it better. You can get a complete list of build targets and other make targets by doing:

```
$ make help
 List of all valid targets in this project:
 ------------------------------------------------------------
 all          : By default generate all lecture
```

```
               code example executables.  Executables
               are compiled into the bin directory.

u01-1        : Unit 01-1 User Defined Functions examples

u01-2        : Unit 01-2 User Defined Data Types examples

u01-3        : Unit 01-3 C/C++ Arrays examples

u02-1        : Unit 02-1 C/C++ Structures

u02-2        : Unit 02-2 C++ Classes

u03-1        : Unit 03-1 Pointer Variables

u03-2        : Unit 03-2 Dynamic Memory

u04-1        : Unit 04-1 Recursion

u05-1        : Unit 05-1 Searching

u05-2        : Unit 05-2 Sorting

u06-1        : Unit 06-1 Analysis of Algorithms

u07-1        : Unit 07-1 Inheritance and Composition

u07-2        : Unit 07-2 Overloading

u07-3        : Unit 07-3 Templates

u09-1        : Unit 09-1 Linked List Fundamentals

u10-1        : Unit 10-1 Implementaiton of Stacks

u10-2        : Unit 10-2 Applications of Stacks

u11-1        : Unit 11-1 Implementation of Queues

u11-2        : Unit 11-2 Applications of Queues

u12-1        : Unit 12-1 Binary Tree Properties

u13-1        : Unit 13-1 Hashing and Dictionaries

u14-1        : Unit 14-1 STL Standard Template Library

readme       : Create repository pdf documentation from
               README.md project markdown file.

format       : Run the code formatter/beautifier by hand if needed

clean        : Remove auto-generated files for a completely
               clean rebuild

help         : Get all build targets supported by this build.
```

Who do I talk to?

Course Instructor: derek dot harter at tamuc dot edu