**CS 255 – Fall 2021 - University of North Alabama**

**Project 1: Fraction Calculator**

**Design Document Due Date: Monday, October 4, 2021 at 11:59 pm  
Project Source Code Due Date: Tuesday, October 5, 2021 at 11:59 pm.**

**Late Policy:** You may submit a design document/project late for up to 24 hours after the deadline with an automatic 20% reduction from the total possible points. You will have a 30-minute grace period to account for any time discrepancies. No work will be accepted after the late date.

**Design Plan Submission:** Upload one file to Canvas, for this project, name the file in the following format:

LastFirst-255-P1-Design.XXX, Example: JenkinsJanet-255-P1-Design.pdf (or word doc is fine). Be sure to email yourself a copy and a timestamp so you will have a record of exactly what you upload to Canvas.

Source Code Submission: Upload one file to Canvas, for this project, name the file in the following format:

LastFirst-255-P1.cpp, Example: JenkinsJanet-255-P1.cpp. Be sure to email yourself a copy and a timestamp so you will have a record of exactly what you upload to Canvas. This will be your driver file.

**Compiler:** I will compile your project in Dev C++.

**Problem** For this project, you will use the Fraction class I have given you (Fraction.h and Fraction.cpp) to create and use Fractions in your driver. Your main program will use the Fraction data type to create a calculator. This calculator should give the user the ability to specify operations and input operands (Fractions). **Your program should make use of all methods and friend functions of the Fraction class. Do not change the Fraction class in any way.**

**Design Plan:** For this project you will required to present a design plan. The design plan will be a typed document and will consist of the following items:

1. Requirements List:

A specific list of general Fraction calculator requirements (something a non-programmer could understand). A requirement should be testable/verifiable in some way.

*Example: The application will display a random Fraction, ask the user to reduce the Fraction, then give an appropriate message of “Correct Reduction”, “Incorrect Reduction”, or “Incorrect Reduction, but Equivalent”.*

1. Function List:

A list of functions that will be used in your project

Each function should have the following 2 items

1. The commented function header in the following format

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//Function Name:

//Brief Function Description

//Return Value:

//Incoming Parameters:

//Outgoing Parameters:

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. Projected time for completion
2. Menu Demo:

Display of a sample menu(s)

1. Test Cases

You should at least have a table of test cases for each requirement and/or function you are testing. This should be a table with headers: “Test Case”, “Result Expected”, “Result Received”, and “Date/Time Tested”. The length of each table will depend upon the job of the function. This testing plan may change and should be updated as you progress. For this submission, just use your plan and any test cases already tested at the time of submission.

**Comments:**

File Header: You should have a header for each file that contains your name, the name of the file, the description of the file (for this project, you will include the project description), the course number, and the due date.

Throughout: Comment major segments of code. Comment any identifiers that do not clearly identify a variable or constant. Comment formulas. Comment any outside sources.

Functions: Use the format listed in the design plan.

**Other notes:**

Do not use global variables.

Keep data members private.

Do not change the Fraction class without prior permission.

You should design your code in such a way that it is reusable. Make appropriate use of functions and reuse existing code where possible.