For all future communications, please use the following SUBJECT LINE:

CIS 25 Fall 2023 YourName : Needs/Questions

Without the above SUBJECT LINE, your emails will be deemed as SPAM/Phishing/Virus and will be deleted.

For all homework submission, please use the following SUBJECT LINE:

cis25Fall2023YourNameHwNumber.cpp

where **Number** is replaced by the homework number, i.e., 1, 2, 3, etc.

Without the above SUBJECT LINE, your submission emails will be rejected, your homework is considered as not submitted.

Turn In:

1. **Exercise #1** – Due on **Tuesday, December 12, 2023 by 11:00 pm as Email Submission**

Homework Due Dates and Consideration:

Students must submit their homework by the given date and time (e.g., Friday on or before 11:00 pm). The homework submissions will be through emailing the work (programs and others) as attachments based on the specified and required formats and structures— only `*.h`, `*.cpp`, no Zip File and no other file formats such as PDF, Word, etc. as you will get zero0) for these non-coding files!

If the homework submission is after 11:00 pm on the given date, then the homework submission is late. A late homework submission within 24 hours will get a 50 % penalty. Between 24 hours and 48 hours late, the penalty will be 75%. After 48 hours late, the submission will not count regardless of the reasons, and the student will receive zero.

Note that by submitting work through emails, there are time stamps for the emails. If you disagree with being marked late or not graded, you can **forward** a copy of your original email submission (as proof of what and when you emailed) for consideration.

There will be about 5 to 6 assignments with specified dues dates. Note that assignments may have 6-day time, 10-day time, or 2+ weeks to submit. Every student must start homework as soon as it is available on Canvas—students are responsible with checking Canvas for class information from the Canvas Homepage, Modules, Quizzes, etc.

Assignment hints may be presented in classes with code samples posted on Canvas. everyone will have the same amount of time to submit regardless of the reasons—work on homework/assignments early.

Homework formats/conventions/styles are explained in class or provided through Canvas. Students must follow the coding formats, conventions, and styles to obtain full credit for all assignments. More information will be available through code demonstration in class — I do coding LIVE during class meetings and discussions, and I use only Visual Studio IDE!

a) For each exercise, a package must be generated to include the following items:
   - Copy of your driver source file (C++ program)—your source file MUST BE NAMED as **`cis25Fall2023FirstLastDriverHw4.cpp`**
   - In addition to the above driver, copies of four (4) other C++ files (i.e., `*.h` header files and the companion `*.cpp` files with proper naming, convention, style, and format as always) should be as follows,

`fractionFirstL.h`	**`fractionFirstL.cpp`**
`fractionUtilityFirstL.h`	**`fractionUtilityFirstL.cpp`**
`hw4UtilityFirstL.h`	**`hw4UtilityFirstL.cpp`**

   - Note! Replace "**YourName**" with your own full-name; i.e., **FirstnameLastName**.
   - Copy of output (copy and paste to the end of your program as **`PROGRAM_OUTPUT`** comment block)
   - Copy of **`Logic_Code_Output_Issues`** comment block (as a separate comment block) after the **`PROGRAM_OUTPUT`**.

b) Emailing each package as follows,
   - One email message for each exercise.
   - The SUBJECT line of the message should have the following line:

     `cis25Fall2023YourNameDriverHw4.cpp`

   - Attaching the source file that was created in part a).

   Note 0! "YourName" means FirstnameLastname—no abbreviation!

2. **Reminder!**

   Getting the program to work *is not enough to earn full credit*. **Your program must run correctly and follow all proper conventions and consistent styles as explained in class, and produce the exact textual information/display as shown/required (of course, with appropriate user input data/values for each run/selection!) to receive credit accordingly.**

   Also, your program must work with all reasonable data sets or patterns.

   Again, writing code is not just the code works. It also involves care, patience, coding idioms + forms, and other reminders. Please see the posted code written in class and the coding convention CPP file.

3. You will get zero (0) points if your code does not compile! Please make sure that you compile your code frequently and correctly throughout the working session. Please check and run your submission again exactly as you just submitted.
4. **Q.E.D.**

****

More Notes!

- **All necessary and required menu(s) must be the combination of `do-while` and `switch` Structures.**
- **You are only allowed to use `cout` and `cin` from the `iostream` header.**
- **You are not allowed to use any other classes or syntax structures that are not introduced in class and class meetings — please confirm with the instructor if you have any doubt!**
- **You are not allowed to use `class`es and functions written by someone else.**
- **You must write all classes and functions yourself before using them in the required class work.**
- **All `class` and `type` names must have your first in full and last names initial appended.**
- **All local variables must be declared at the top of its function.**
- Except for the member data, member functions, function arguments and local variables within member functions, and indices of `i`, `j`, `k`, etc., all other variables must have the initials of your first name and last name added to the end of the variable names.

  For examples,

```
int usrInputFL;
int digitCountFL;
int absValueFL;
int tmpFL;
```

- **All stand-alone function names must have the initials of your Firstname and Lastname appended at the end.**
- **All filenames must have your complete Firstname and Lastname appended as required.**
- **The values of the elements from the created array, it any, MUST NOT be changed/modified at all during the analysis steps and process unless they are required to be updated or changed based on the specified tasks.**

Reminder!

- In your program, no GLOBAL DATA are allowed, and you must write all needed functions (no library functions are allowed – Except for **`cin`** and **`cout`** and their functions/manipulators).
- No use of `extern` modifier!
- All user-created objects must be dynamic — creating through operator `new`.
- All dynamic allocations must be released/deleted properly — through operator `delete`!
- One `new` ←→ One proper `delete` — syntax errors or/and heavy penalties if this is not followed!
- Smart pointers are created differently and used differently; they are not used in this class — please do not use smarter pointers!

- Again, writing code is not just the code works. It also involves care, patience, coding idioms + forms, and other reminders. Please see the posted code written in class and the explanation of coding convention CPP file.

++++++++++

Exercise 1 – Due Tuesday, December 12, 2023 by 11:00 pm through Email Submission

A. Update the **`Fraction`** class given in the Lecture notes or as discussed in class meetings as follows,

1. Add your FIRST NAME and the initial of your last name to the name **`Fraction`** and use this as your updated class. For examples, if your first name is **`First Last`** then update the class name to be **`FractionFirstL`**.
2. Add and update all class constructors for your **`FractionFristL`** class to handle the initialization appropriately.

   There must be as least 3 constructors of

   (i) Default; and

   (ii) Copy; and

   (iii) **`FractionFirstL(int n, int d)`**

3. Provide a destructor with a confirmation when removing the object (i.e., "**`Calling ~FractionFirstL()`**").
4. Provide `get()/set()` **member** functions for each private member data. That means

   `getNum()` `getDenom()`

   `setNum()` `setDenom()`

B. Provide the following **member** functions,

a. A member function `isNumPalindrome()`, which is a predicate returning true/false if the member `num` of the `Fraction` object is a Palindrome or not; and

b. A member function `isDenomPalindrome()`, which is a predicate returning true/false if the member `num` of the `Fraction` object is a Palindrome or not; and

c. A `gcd()` helper involves in making a proper `Fraction` object.

```
/**
 * Program Name: fractionFirstL.h
 * Discussion:   File #1 -
 *                 Specification File
 *                   for your FractionFirstL class
 * Written By:   First Last
 * Date:         2023/12/xx
 */

#ifndef FRACTIONFIRSTL_H
#define FRACTIONFIRSTL_H

// Include/Header File(s)
#include <iostream>
using namespace std;

// Rules/Constraints for FractionFirstL object
```

```
//   Rule #1:
//     denom cannot be zero!
//   Rule #2:
//     No common factor berween num and denom
//   Rule #3:
//     Negativity is imposed on num

class FractionFirstL {
public:
    FractionFirstL();
    FractionFirstL(const FractionFirstL&);
    FractionFirstL(int, int);

    ~FractionFirstL();

    int getNum(void) const;
    void setNum(int);

    int getDenom(void) const;
    void setDenom(int);

    void update(int, int);
    void print(void) const;

    bool isNumPalindrome(void) const;
    bool isDenomPalindrome(void) const;

    friend ostream& operator<<(ostream&,
        const Fraction&);
private:
    int num;
    int denom;

    int gcdA(int, int) const;};

#endif


/**
 * Program Name: fractionFirstL.cpp
 * Discussion:   File #2 -
 *                 Implementation File
 * Written By:   First Last
 * Date:         2023/12/xx
 */

// Include/Header File(s)
#include <iostream>
#include "fractionFirstL.h"
using namespace std;

// Function Definitions
```

C. Provide the following stand-alone (non-member) functions in the `fractionUtility` files,

   a. A predicate `isPalindromeFL()`, to check if the `FractionFirstL` object is a Palindrome; and

   b. A function `displayCommonPalindromeDigitFL()` to display the information from the Palindrome `FractionFirstL` object.

   c. Other functions as needed.

```
/**
 * Program Name: fractionUtilityFirstL.h
 * Discussion:   File #3 -
 *                 Specification File
 *                   Utility
```

```
 * Written By:   First Last
 * Date:         2023/11/xx
 */

#ifndef FRACTIONUTILITYFIRSTL_H
#define FRACTIONUTILITYFIRSTL_H

// Include/Header File(s)
#include <iostream>
#include "fractionFirstL.h"
using namespace std;

// Required Function Protyotypes

bool isPalindromeFL(const Fraction*);
void displayCommonPalindromeDigitFL(const Fraction*);
void createFractionFL(Fraction*& frRef);
void updateFractionFL(Fraction*& frRef);

// Other functions as needed

#endif
```

```
/**
 * Program Name: fractionUtilityFirstL.cpp
 * Discussion:   File #6 -
 *                 Implementation File
 *                   Fraction Utility
 * Written By:   First Last
 * Date:         2023/12/xx
 */

// Include/Header File(s)
#include <iostream>
#include "fractionFirstL.h"
#include "fractionUtilityFirstL.h"
using namespace std;

// Function Definitions Here
```

D. Provide the following stand-alone functions in the `hw4Utility` files,

   a. A proper `initSubMenuFL()` to create or update the required `FractionFirstL` object.

   b. An appropriate `runMenuHw4FL()` function to produce the required output as displayed below; and

   c. Other functions as needed.

```
/**
 * Program Name: hw4UtilityFirstL.h
 * Discussion:   File #5 -
 *                 Specification File
 *                   HW #4 Utility
 * Written By:   First Last
 * Date:         2023/12/xx
 */

#ifndef HW4UTILITYFIRSTL_H
#define HW4UTILITYFIRSTL_H

// Include/Header File(s)
#include <iostream>
#include "fractionFirstL.h"
#include "fractionUtilityFirstL.h"
using namespace std;
```

```
// Required functions

void displayCodingStatementFL(void);
void displayClassInfoHw4FL(void);
void runMenuHw4FL(void);
void initSubMenuFL(FractionFirstL*&);

// Other functions as needed

#endif

/**
 * Program Name: hw4UtilityFirstL.cpp
 * Discussion:   File #6 -
 *                 Implementation File
 *                   Utility
 * Written By:   First Last
 * Date:         2023/12/xx
 */

// Include/Header File(s)
#include <iostream>
#include "fractionFirstL.h"
#include "fractionUtilityFirstL.h"
#include "hw4UtilityFirstL.h"
using namespace std;

// Function Definitions Here
```

E. Run a driver/program named as **`cis25Fall2023YourNameDriverHw4.cpp`** and record the output shown below.

```
/**
 * Program Name: cis25Spring2023FirstLastDriverHw4.cpp
 * Discussion:   File #7 -
 *                 Application Driver
 * Written By:   First Last
 * Date:         2023/12/10
 */

// Include/Header File(s)
#include <iostream>
#include "fractionFirstL.h"
#include "fractionUtilityFirstL.h"
#include "hw4UtilityFirstL.h"
using namespace std;

// Application Driver
int main() {

    displayCodingStatementFL();
    displayClassInfoHw4FL();

    runMenuHw4FL();

    return 0;
}

/** PROGRAM OUTPUT

*/

/** Logic_Code_Output_Issues
No Comments! <-- Replace if having issues
*/
```

b) The setup code has been posted. Students should download and use this setup code. Additional work and details must be provided for some of the functions and any other additional required code and or setup (that you see or may have).

c) You must update the code to meet the naming requirements. For examples, the class `Fraction` should be updated to be `FractionFirstL`, replacing `FirstL` with your own name information, etc.

d) The output screen should have the following lines displayed before any other display or input can be seen,

```
We write code to manipulate data (which are
provided by the user) to produce the
required outcome in the most efficient way!

CIS 25 - C++ Programming
Laney College
Firstname Lastname

Information --
  Assignment:               HW #4
  Implemented by:           Firstname Lastname
  Required Submission Date: 2023/12/12
  Actual Submission Date:   2023/12/__

**************************************************
*                   MENU - HW #4                 *
*  1. Setting Up Fraction through initSubMenuFL() *
*  2. Calling isPalindromeFL()                    *
*  3. Calling displayCommonPalindromeDigitFL()    *
*  4. Printing Current Fraction                   *
*  5. Quit                                        *
**************************************************
Select an option (use integer value only): 6

WRONG OPTION!

**************************************************
*                   MENU - HW #4                 *
*  1. Setting Up Fraction through initSubMenuFL() *
*  2. Calling isPalindromeFL()                    *
*  3. Calling displayCommonPalindromeDigitFL()    *
*  4. Printing Current Fraction                   *
*  5. Quit                                        *
**************************************************
Select an option (use integer value only): 2

Using isPalindromeFL() Option -

  Not a proper option as there is no Fraction!

**************************************************
*                   MENU - HW #4                 *
*  1. Setting Up Fraction through initSubMenuFL() *
*  2. Calling isPalindromeFL()                    *
*  3. Calling displayCommonPalindromeDigitFL()    *
```

```
*  4. Printing Current Fraction                    *
*  5. Quit                                         *
**************************************************
Select an option (use integer value only): 1

INITIALIZING Option -

Calling initSubMenuFL() -

  ******************************
  * initSubMenu - One Fraction *
  *  1. Creating               *
  *  2. Updating               *
  *  3. Returning              *
  ******************************
  Select an option (integer only): 2

  Not a proper option as there is no Fraction!

  ******************************
  * initSubMenu - One Fraction *
  *  1. Creating               *
  *  2. Updating               *
  *  3. Returning              *
  ******************************
  Select an option (integer only): 5

  WRONG OPTION!

  ******************************
  * initSubMenu - One Fraction *
  *  1. Creating               *
  *  2. Updating               *
  *  3. Returning              *
  ******************************
  Select an option (integer only): 1

  Creating 1 NEW Fraction object --

  Calling createFractionFL()!

    Enter num: 5
    Enter denom: -959

    One Fraction of
      Address : 0041FDD4
        num : -5
        denom : 959
    has just been created/built!

  ******************************
  * initSubMenu - One Fraction *
  *  1. Creating               *
  *  2. Updating               *
  *  3. Returning              *
  ******************************
```

```
  Select an option (integer only): 1

  Please update or return!

  ****************************
  * initSubMenu - One Fraction *
  *  1. Creating               *
  *  2. Updating               *
  *  3. Returning              *
  ****************************
  Select an option (integer only): 3

  Returning to previous menu!

**************************************************
*                  MENU - HW #4                  *
*  1. Setting Up Fraction through initSubMenuFL() *
*  2. Calling isPalindromeFL()                    *
*  3. Calling displayCommonPalindromeDigitFL()    *
*  4. Printing Current Fraction                   *
*  5. Quit                                        *
**************************************************
Select an option (use integer value only): 4

PRINTING Option -

  Address : 0041FDD4
    num : -5
    denom : 959

**************************************************
*                  MENU - HW #4                  *
*  1. Setting Up Fraction through initSubMenuFL() *
*  2. Calling isPalindromeFL()                    *
*  3. Calling displayCommonPalindromeDigitFL()    *
*  4. Printing Current Fraction                   *
*  5. Quit                                        *
**************************************************
Select an option (use integer value only): 2

Using isPalindromeFL() Option -

  The current Fraction is a Palindrome!

**************************************************
*                  MENU - HW #4                  *
*  1. Setting Up Fraction through initSubMenuFL() *
*  2. Calling isPalindromeFL()                    *
*  3. Calling displayCommonPalindromeDigitFL()    *
*  4. Printing Current Fraction                   *
*  5. Quit                                        *
**************************************************
Select an option (use integer value only): 3

Calling displayCommonPalindromeDigitFL() Option -
```

```
  There is/are 1 common digit(s) of
    5

  The largest common Palindrome digit: 5

**************************************************
*                   MENU - HW #4                 *
*  1. Setting Up Fraction through initSubMenuFL() *
*  2. Calling isPalindromeFL()                    *
*  3. Calling displayCommonPalindromeDigitFL()    *
*  4. Printing Current Fraction                   *
*  5. Quit                                        *
**************************************************
Select an option (use integer value only): 1

INITIALIZING Option -

Calling initSubMenuFL() -

  ******************************
  * initSubMenu - One Fraction *
  *  1. Creating               *
  *  2. Updating               *
  *  3. Returning              *
  ******************************
  Select an option (integer only): 1

  Please update or return!

  ******************************
  * initSubMenu - One Fraction *
  *  1. Creating               *
  *  2. Updating               *
  *  3. Returning              *
  ******************************
  Select an option (integer only): 2

  Updating an EXISTING Fraction object -

  Calling updateFractionFL()!

    Enter num: 156810000
    Enter denom: 49914173

    The Fraction object at 0041FDD4 has been updated as
      num : 156810000
      denom : 49914173

  ******************************
  * initSubMenu - One Fraction *
  *  1. Creating               *
  *  2. Updating               *
  *  3. Returning              *
  ******************************
  Select an option (integer only): 3
```

```
  Returning to previous menu!

**************************************************
*                   MENU - HW #4                 *
*  1. Setting Up Fraction through initSubMenuFL() *
*  2. Calling isPalindromeFL()                    *
*  3. Calling displayCommonPalindromeDigitFL()    *
*  4. Printing Current Fraction                   *
*  5. Quit                                        *
**************************************************
Select an option (use integer value only): 4

PRINTING Option -

  Address : 0041FDD4
    num : 156810000
    denom : 49914173

**************************************************
*                   MENU - HW #4                 *
*  1. Setting Up Fraction through initSubMenuFL() *
*  2. Calling isPalindromeFL()                    *
*  3. Calling displayCommonPalindromeDigitFL()    *
*  4. Printing Current Fraction                   *
*  5. Quit                                        *
**************************************************
Select an option (use integer value only): 2

Using isPalindromeFL() Option -

  The current Fraction is not a Palindrome!

**************************************************
*                   MENU - HW #4                 *
*  1. Setting Up Fraction through initSubMenuFL() *
*  2. Calling isPalindromeFL()                    *
*  3. Calling displayCommonPalindromeDigitFL()    *
*  4. Printing Current Fraction                   *
*  5. Quit                                        *
**************************************************
Select an option (use integer value only): 3

Calling displayCommonPalindromeDigitFL() Option –

  The current Fraction is not a Palindrome!

**************************************************
*                   MENU - HW #4                 *
*  1. Setting Up Fraction through initSubMenuFL() *
*  2. Calling isPalindromeFL()                    *
*  3. Calling displayCommonPalindromeDigitFL()    *
*  4. Printing Current Fraction                   *
*  5. Quit                                        *
**************************************************
Select an option (use integer value only): 1
```

```
INITIALIZING Option -

Calling initSubMenuFL() -

  ******************************
  * initSubMenu - One Fraction *
  *  1. Creating               *
  *  2. Updating               *
  *  3. Returning              *
  ******************************
  Select an option (integer only): 1

  Please update or return!

  ******************************
  * initSubMenu - One Fraction *
  *  1. Creating               *
  *  2. Updating               *
  *  3. Returning              *
  ******************************
  Select an option (integer only): 2

  Updating an EXISTING Fraction object -

  Calling updateFractionFL()!

    Enter num: 1551
    Enter denom: 5491945

    The Fraction object at 0041FDD4 has been updated as
      num : 1551
      denom : 5491945

  ******************************
  * initSubMenu - One Fraction *
  *  1. Creating               *
  *  2. Updating               *
  *  3. Returning              *
  ******************************
  Select an option (integer only): 3

  Returning to previous menu!

**************************************************
*                  MENU - HW #4                  *
*  1. Setting Up Fraction through initSubMenuFL() *
*  2. Calling isPalindromeFL()                    *
*  3. Calling displayCommonPalindromeDigitFL()    *
*  4. Printing Current Fraction                   *
*  5. Quit                                        *
**************************************************
Select an option (use integer value only): 4

PRINTING Option -

  Address : 0041FDD4
```

```
    num : 1551
    denom : 5491945

**************************************************
*                   MENU - HW #4                 *
*  1. Setting Up Fraction through initSubMenuFL() *
*  2. Calling isPalindromeFL()                    *
*  3. Calling displayCommonPalindromeDigitFL()    *
*  4. Printing Current Fraction                   *
*  5. Quit                                        *
**************************************************
Select an option (use integer value only): 2

Using isPalindromeFL() Option –

  The current Fraction is a Palindrome!

**************************************************
*                   MENU - HW #4                 *
*  1. Setting Up Fraction through initSubMenuFL() *
*  2. Calling isPalindromeFL()                    *
*  3. Calling displayCommonPalindromeDigitFL()    *
*  4. Printing Current Fraction                   *
*  5. Quit                                        *
**************************************************
Select an option (use integer value only): 3

Calling displayCommonPalindromeDigitFL() Option –

  There is/are 2 common digit(s) of
    1
    5

  The largest common Palindrome digit: 5

**************************************************
*                   MENU - HW #4                 *
*  1. Setting Up Fraction through initSubMenuFL() *
*  2. Calling isPalindromeFL()                    *
*  3. Calling displayCommonPalindromeDigitFL()    *
*  4. Printing Current Fraction                   *
*  5. Quit                                        *
**************************************************
Select an option (use integer value only): 5

The Fraction is at
  Address : 0041FDD4
    num : 1551
    denom : 5491945

Calling ~FractionFirstL()

The Fraction is now removed!

Having fun ...!
```

At least, your program should have and use the following functions,

```
displayCodingStatementFL()
displayClassInfoFL()
runMenuHw4FL()
```

where YourName must be replaced by your first name initial and your last name initial . For examples,

If your name is **First Last** then the initial of FL should be usedthroughout all of your work/code as mentioned for variable names and function names appropriately as mentioned above.

The sample run will have the options and values selected by the user. While testing, you must run your program to produce the required output.

(4) Save the work with a driver of

```
cis25Fall2023YourNameDriverHw4.cpp
```

The above output should be copied and added to the end of the code in the PROGRAM_OUTPUT comment block. No manual manipulation to the output is allowed; you will get zero (0) for the whole work if any manual manipulation is found!

(5) Email the source code (your program) above using the SUBJECT LINE of

```
cis25Fall2023YourNameDriverHw4.cpp
```

Again, THE Notes!

- **All necessary and required menu(s) must be the combination of `do-while` and `switch` Structures.**
- **You are only allowed to use `cout` and `cin` from the `iostream` header.**
- **You are not allowed to use any other classes or syntax structures that are not introduced in class and class meetings — please confirm with the instructor if you have any doubt!**
- **You are not allowed to use `class`es and functions written by someone else.**
- **You must write all classes and functions yourself before using them in the required class work.**
- **All `class` and `type` names must have your first in full and last names initial appended.**
- **All local variables must be declared at the top of its function.**
- Except for the member data, member functions, function arguments and local variables within member functions, and indices of `i`, `j`, `k`, etc., all other variables must have the initials of your first name and last name added to the end of the variable names.

  For examples,

```
int usrInputFL;
int digitCountFL;
int absValueFL;
```

```
int tmpFL;
```

- **All stand-alone function names must have the initials of your Firstname and Lastname appended at the end.**
- **All filenames must have your complete Firstname and Lastname appended as required.**
- **The values of the elements from the created array, it any, MUST NOT be changed/modified at all during the analysis steps and process unless they are required to be updated or changed based on the specified tasks.**

Reminder!

- In your program, no GLOBAL DATA are allowed, and you must write all needed functions (no library functions are allowed – Except for **`cin`** and **`cout`** and their functions/manipulators).
- No use of `extern` modifier!
- All user-created objects must be dynamic — creating through operator `new`.
- All dynamic allocations must be released/deleted properly — through operator `delete`!
- One `new` ←→ One proper `delete` — syntax errors or/and heavy penalties if this is not followed!
- Smart pointers are created differently and used differently; they are not used in this class — please do not use smarter pointers!
- Again, writing code is not just the code works. It also involves care, patience, coding idioms + forms, and other reminders. Please see the posted code written in class and the explanation of coding convention CPP file.