

CSCI 2170 Example Test 3 questions

1. Quiz 2, Quiz3, and Quiz 4 questions

2. Closed lab program for closed labs 13, 14, 15 and 16.

3. ADT related questions:

A complex number consists of two components: the real component and the imaginary component. An example of a complex number is $2+3i$, where 2 is the real component and 3 is the imaginary component of the data. Define a class `MyComplexClass`. It has two data values of float type: **real** and **imaginary**.

This class has the following member functions:

- A **default constructor** that assigns 0.0 to both its real and imaginary data members;
- The **value constructor** that assigns client supplied (real and imaginary) values to the real and imaginary data members;
- The copy constructor
- A **member function "SetValues"** that assigns client supplied values to the real and imaginary data members; (This is not a constructor);
- A member function **"GetReal"** that returns the real component of the number;
- A member function **"Display"** that outputs the complex number in the form " $a + bi$ " on screen, where a and b are the real and imaginary components.
- A **member function "EqualTo"** that compares two complex numbers. It returns true if they are the same, and returns false if they are different. Two complex numbers are considered the same if the real components of the two values are the same and the imaginary components of the two values are also the same.
- **Overloaded == operator** that compares two complex numbers. Two complex numbers are equal if they have the same real value and the same imaginary value.
- The **overloaded << (insertion) operator**. The operator outputs to an output stream a complex number in the form:
real + imaginary i
- The **overloaded >> (extraction) operator**. The operator reads two float values from the input stream and assign to the real and imaginary values of a complex object.
- Create array of complex objects
- Call methods to compare two complex objects
- Call method to display the complex objects
- Write user defined functions that pass complex objects by value or by reference

You are required to:

(a) Write the complete header file for `MyComplexClass`; (you are not required to write comments)

(b) Write the complete implementation file for `MyComplexClass`.

(c) Write the client program to:

- Create two objects of `MyComplexClass`. One objects should be created using the default constructor, and the other with the value constructor;
- Use overloaded `>>` operator to read the values of the first complex number;
- Display the first complex numbers using the **Display** member function;
- Use overloaded `<<` operator to display the second complex number;
- Apply **EqualTo** function to compare the two complex numbers and output appropriate messages concerning whether the two numbers are the same or not.
- Declare the third complex number as a copy of the second complex number
- Write a user defined functions to

- Add two complex numbers. For two complex numbers: $a+bi$ and $c+di$, the addition of the two numbers is: $(a+c) + (b+d)i$
- Declare an array of 20 complex numbers
- Write a user defined functions to
 - Display each of these 20 complex number

4. Write C++ client program code segments using unsorted or sorted list class as discussed in class

Write a C++ program to read in a number of integer values from a data file and add to (insert into) the list. Then it calls the user-defined function “FindSum” to compute and display the sum of these values.

```
#include <iostream>
// include additional file here
```

```
using namespace std;
// declare user defined function here
```

```
int main() {
```

```
    // declare a list class object here
```

```
    // read values from the data file and add to the list, til the end of the file is reached
```

```
    // call “FindSum” function to compute and display the sum of these values.
```

```
    return 0;
}
```

```
// define the function “FindSum” below
```