CS 132 Data Structure
2017 Spring Quarter

Homework Assignment #13

Due : April 24 (Monday)

Each program should have a section of comments at the top. The comments should contain your name and your partners' name, the date the program was written, the assignment number, and how long you spent on this assignment.
Only some parts of homework assignment will be graded (Not whole homework). You have to email me (jsuk@greenriver.edu) for C++ codes (cpp files) and submit a hard copy of your homework assignments before the due - April 24 (Monday) at the beginning of class.

♣ Project ;

1. Make a group (maximum is 4.) Your group may keep same members.
2. Keep the previous project program (basic idea) and edit it for improving the program from CS& 131 course.
3. Design a class for storing data.
4. Define appropriate a constructor and a destructor.
5. Make appropriate accessor and mutator member functions.
6. Confirm the plan of your group with the instructor.

♣ Textbook Problems ;

○ Chapter 13

Short Answer

4. Look at the following function header for a member function.

```
void Circle::getRadius()
```

What is the name of the function?
What class is the function a member of?

10. What is a constructor? What is a destructor?

Fill-in-the-Blank

24. The default access specification of class member is ________.

31. A(n) ___________ is automatically called when an object is created.

36. A(n) ___________ is a member function that is automatically called when an object is destroyed.

Algorithm Workbench

43. Write a class declaration named **Circle** with a private member variable named **radius**. Write set and get functions to access the **radius** variable, and a function named **getArea** that returns the area of the circle. The area is calculated as

```
3.14159 * radius * radius
```

Program Challenge

1. Date

Design a class called **Date**. The class should store a date in three integers: **month**, **day**, and **year**. There should be member functions to print the date in the following forms:

```
12/25/2014
December 25, 2014
25 December 2014
```

Demonstrate the class by writing a complete program implementing it.
Input Validation: Do not accept values for the day greater than 31 *or less than* 1. *Do not accept values for the month greater than* 12 *or less than* 1.

2. **Employee** Class

Write a class named **Employee** that has the following member variables:

- **name.** A string that holds the employee's name.
- **idNumber.** An **int** variable that holds the employee's ID number.
- **department.** A string that holds the name of the department where the employee works.
- **position.** A string that holds the employee's job title.

The class should have the following constructors:

- A constructor that accepts the following values as arguments and assigns them to the appropriated member variable: employee's name, employee's ID number, department, and position.
- A constructor that accepts the following values as arguments and assigns them to the appropriate member variables: employee's name and ID number. The **department** and **position** fields should be assigned an empty string (" ").
- A default constructor that assign empty strings (" ") to the **name**, **department**, and **position** member variables, and 0 to the **idNumber** member variable.

Write appropriate mutator functions that store values in these member variables and accessor functions that return the values in these member variables. Once you have written the class, write a separate program that creates three **Employee** object to hold the following data. The program should store this data in the three objects and then display the data for each employee on the screen.

Name	**ID Number**	**Department**	**Position**
Susan Meyers	47899	Accounting	Vice President
Mark Jones	39119	IT	Programmer
Joy Rogers	81774	Manufacturing	Engineer

4. Personal Information Class
Design a class that holds the following personal data: name, address, age, and phone number. Write appropriate accessor and mutator functions. Demonstrate the class by writing a program that creates three instances of it. One instance should hold your information, and the other two should hold your friends' or family members' information.

10. Number Array Class
Design a class that has an array of floating-point numbers. The constructor should accept an integer argument and dynamically allocate the array to hold that many members. The destructor should free the memory held by the array. In addition, there should be member functions to perform the following operations:

- Store a number in any element of the array
- Retrieve a number from any element of the array
- Return the highest value stored in the array
- Return the lowest value stored in the array
- Return the average of all the numbers stored in the array

Demonstrate the class in a program.

12. Coin Toss Simulator
Write a class named **Coin**. The **Coin** class should have the following member variable:

- A **string** named **sideUp**. The **sideUp** member variable will hold either ”heads” or ”tails” indicating the side of the coin that is facing up.

The **Coin** class should ave the following member functions:

- A default constructor that randomly determines the side of the coin that is facing up (”head” or ”tails”) and initializes the **sideUp** member variable accordingly.
- A **void** member function named **toss** that simulates the tossing of the coin. When the **toss** member function is called, it randomly determines the side of the coin that is facing up (”heads” or ”tails”) and sets the **sideUp** member variable accordingly.
- A member function named **getSideUp** that returns the value of the **sideUp** member variable.

Write a program that demonstrates the **Coin** class. The program should creates an instance of the class and display the side that is initially facing up. Then, use loop to toss the coin 20 times. Each time the coin is tossed, display the side that is facing up. The program should keep count of the number of times heads is facing up and the number of times tails is facing up, and display those values after the loop finishes.

★ You have to make an individual cpp file for each programming challenges and project program.
★ Don’t forget your comments about your program.