Extra Credit Assignment (up to 20 points; due Friday, 12/15/2023, at 11:59:00 P.M.)

For all Programs:

To get in the habit of writing pseudocode write the simple pseudocode for these programs. Put your pseudocode in comments at the top of your programs, as well as many lines within the code necessary to describe what's going on in the program. PSEUDOCODE IS REQUIRED FOR ALL PROGRAMS.

For each program make sure and include the following comments at the top (do this on all homework assignments from now on – this is required):

```
//Your Name
//CS 1, Section #ABC
//Assignment #X, Problem #Y
//Summary of the program
```

For this homework you have three programs so you will be emailing 2 program files. These programs address Chapters 13 and 14.

Employee Info

1. (10 points) Write a class named Employee that has the following member variables:

   - name: The name attribute holds an employee's first and last name
   - idNumber: The idNumber attribute is a string variable that holds an employee's ID number
   - department: The department attribute holds the name of the employee's department
   - position: The position attribute holds the name of the employee's job title
   - yearsWorked: This attribute holds the number of years the employee has worked at the company

   The class should have the following overloaded constructors:

   - A constructor that accepts the following values as arguments and assigns them to the appropriate data members (attributes): employee's name and employee's ID number, employee's department, employee's position, and the number of years the employee has worked at the company.

- A constructor that accepts the following values as arguments and assigns them to the appropriate data members (attributes): employee's name and employee's ID number. The department and position attributes should be initialized to the empty string ("") and the yearsWorked attribute should be initialized to zero.

- A default constructor (accepts no arguments) that initializes the name, idNumber, department, and position attributes to the empty string (""). The yearsWorked attribute should be initialized to zero.

Write Get and Set methods for each attribute: name, idNumber, department, position, and yearsWorked.

Do not allow the yearsWorked attribute to be set to values less than zero. If an attempt is made to set yearsWorked to less than zero, do not set the attribute and communicate the problem to the calling program.

Remember that the class declaration (.h file) will contain the class attributes, function prototypes, and the function definitions. Name the class declaration file employee.h.

Next create a program to utilize the Employee class you created in the following manner:

- Declare 3 Employee objects
    - One with all 5 args included
    - One with only the first two args included
    - One with no args.

Use set functions to set the missing values of the partially constructed objects. Demonstrate the error detection of the setyearsWorked function.

Display the data for each of the three employees to the screen in similar to the table below (you do not need to display the lines in the table):

Name	**ID Number**	**Department**	**Position**	**Years Worked**
Jenny Jacobs	JJ8990	Accounting	President	15
Myron Smith	MS7571	IT	Programmer	5
Chris Raines	CR6873	Manufacturing	Engineer	30

The program utilizing the employee class should be in a separate .cpp file. Name this program employeeTest.cpp (don't forget to include the employee.h file).

Stats Class and Rainfall Statistics Program

2. (10 points) Design and create a class named Stats that has an array of 12 doubles as one of its member variables. The values in the array should be set by making 12 calls to a public member function named setValue that accepts two arguments, an integer indicating which value is being provided (the first number, the second number….etc) and a double holding the actual data value.

   In addition to the setValue member function, the class should have the following additional member functions:
   - A default constructor that sets all of the values in the array to zero
   - A getValues function that displays all of the values in the array
   - A getTotal function that returns the total of the 12 values in the array
   - A getAvg function that returns the average of the 12 values in the array
   - A getLargest function that returns the largest value in the array
   - A getSmallest function that returns the smallest values in the array

   Remember that the class specification/declaration (.h file) will contain the member variables, member function prototypes, and the function definitions. Name the class declaration file Stats.h.

   Next create a program that uses the Stats class to hold rainfall data and report annual rainfall statistics. The program should first create a Stats object named rainfall and call its setValue member function to set each of the 12 monthly rainfall totals to a user entered amount. It should then produce an annual rainfall report that shows the total, average, lowest and highest rainfall amounts for the year. Note that the printing of the rainfall report should be done in the client program file not in the Stats class.

   Input Validation: If any amount less than 0 is passed into the setValue function, a default value of 0 should be used in his place.

   The program utilizing the stats class should be in a separate .cpp file. Name this program rainfall.cpp (don't forget to include the Stats.h file).