Lab 2: Review Classes and Objects

CPRG4202 – T. MacDonald

In this lab you and your lab partner will create a new class called **WorkTicket**. Objects of this class could be used in an IT support tracking application to store information about client requests for support.

WorkTicket Class Requirements:

- □ ~~**Attributes**. An object of class WorkTicket has the following attributes. Include a **set** (*mutator*) and **get** (*accessor*) method for each attribute.~~
  - ~~*Work Ticket Number* – A whole, positive number. If a work ticket number is set to a zero or a negative number, an `invalid_argument` exception should be thrown, with an appropriate message.~~
  - ~~*Client ID* – The alpha-numeric code assigned to the client.~~
  - ~~*Work Ticket Date* – The date the ticket was opened, stored as three integer numbers. The day must be between 1 and 31. The month must be between 1 and 12. The year must be 4 digits and in the 21st Century (between 2000 and 2099). An `invalid_argument` exception should be thrown, with an appropriate message if the day, month, or year is set out of range.~~
  - ~~*Issue Description* – A description of the issue the client is having.~~
- □ **Default and parameterized constructor(s)**. If parameters are not specified, set the work ticket number to zero, the work ticket date to 1/1/2000, and all other attributes to empty strings.
- □ **SetWorkTicket()** – a *mutator* method to set all the attributes of the object to the parameters as long as the parameters are valid. ALL of the parameters must be valid in order for ANY of the attributes to change. Validation rules are explained above for work ticket number and date. Client number and Description must be at least one character long. If no problems are detected, return TRUE. Otherwise return FALSE.
- □ **ShowWorkTicket( ).** An *accessor* method to display all the object's attributes neatly in the console window.

Program Requirements:

The purpose of the `main()` function in this program is to demonstrate each of the features of the WorkTicket class.

- □ Create an array of at least three WorkTicket objects.
- □ Use an input loop to allow the user to enter the work ticket information. The input loop should call the SetWorkTicket( ) method.
- □ Then use an output loop to display each of the work tickets in the array.

Things to Explore:

You are welcome to explore beyond the mandatory requirements if you wish. Some suggestions you may be interested in:

- □ **Use a pointer to dynamically allocate objects, or use a standard vector.** You could try this out in your demo program rather than a simple array.
- □ **Including static members in your class.** Consider adding one or more attributes that would be shared by all WorkTicket objects.

General Requirements

- Include an opening comment with your full name, the full names on the student(s) you are working with, the name of the program, the date, and a short description.
- Follow the style guide! Use descriptive names and sensible data-types for variables, constants, arrays, functions, etc. that follow our naming conventions. Use good spacing and make sure braces ({}) are located where they are supposed to be.
- Attach the unzipped source code file(s) (*.cpp, .h*) to the dropbox.