Problem Set 04 - OOP - Part I

Mandatory

For each class below create a header file with the same name and for each C++ program create a file with a name in the format

`main4`*n*`.cpp`

where *n* is the number of the program in the list below.

Tasks:

1. Create the class *Complex* that contains

   - □ A private double field named *real*.
   - □ A private double field named *imaginary*.
   - □ A public default constructor that assigns 1 and 0 to the fields *real* and *imaginary*, respectively.
   - □ A public overloaded constructor that takes a double parameter and assigns the parameter and 0 to *real* and *imaginary*, respectively.
   - □ A public overloaded constructor that takes two double parameters and assigns the parameters to *real* and *imaginary* in order.
   - □ A public copy constructor.
   - □ A public overloaded assignment operator.
   - □ A public empty destructor.
   - □ A public double constant method named `GetReal()` that takes no parameters and returns *real*.
   - □ A public double constant method named `GetImaginary()` that takes no parameters and returns *imaginary*.
   - □ A public void method named `SetReal()` that takes a double parameter and assigns the parameter to *real*.
   - □ A public void method named `SetImaginary()` that takes a double parameter and assigns the parameter to *imaginary*.
   - □ a public string constant method named `ToString()` that takes no parameters and returns a string in the format

     $$\begin{cases} x & \text{if } imaginary = 0 \\ y\,\texttt{i} & \text{if } real = 0 \\ x\ \texttt{+}\ y\,\texttt{i} & \text{if } imaginary > 0 \\ x\ \texttt{-}\ z\,\texttt{i} & \text{if } imaginary < 0 \end{cases}$$

     where x, y, z are the values of *real*, *imaginary*, and the absolute value of *imaginary*, respectively, with one decimal point.
   - □ a friend overloaded ostream operator that displays its output in the same format as `ToString()`.

2. Create a program that

   - □ Initializes 6 *Complex* objects such that one is only real, imaginary, and has a negative imaginary part.
   - □ Displays the objects.
   - □ Creates four new *Complex* objects whose values are the results of addition, subtraction, multiplication, and division operations of the other objects.
   - □ Displays the new objects.

3. Class *Pin* that contains

   - □ A private string field named *pin*.
   - □ A private Boolean field named *view*.
   - □ A public default constructor that assigns `"1234"` and false to *pin* and *view* respectively.
   - □ A public copy constructor.
   - □ A public overloaded assignment operator.
   - □ A public empty destructor.
   - □ A public void method named `SetPin()` that takes a string parameter. It assigns the parameter to *pin* only if the parameter is a string of 4 digits.
   - □ A public void method named `SetView()` that takes a Boolean parameter and assigns the parameter to *view*.
   - □ A public string constant method named `ToString()` that takes no parameters and returns a string in the format

     $$\begin{cases} x & \text{if } view \text{ is true} \\ \texttt{"****"} & \text{if } view \text{ is false} \end{cases}$$

     where x is the value of *pin*.
   - □ A friend overloaded ostream operator that displays its output in the same format as `ToString()`.
   - □ A friend overloaded equivalence operator (`==`) that returns true if the *pin* fields of its parameters are the same; otherwise, it returns false.

4. Create a program that

   - □ Creates a *Pin* object and assigns it a random pin (four-digit string).
   - □ Continually prompts the user to enter a pin until it matches the object (this requires creating a new object).
   - □ Displays the object.