Practice 02 (01-10-2020)

1. Create a class that imitates part of the functionality of the basic/ primitive data type int. Call the class Int (note different capitalization). The only data member in this class is an integer variable. Include member functions:

   - Non-parameterized constructor, to initialize data member with 0
   - Parameterized constructor to initialize data member with the parameter passed
   - display function to display it (it looks just like an int)
   - add function to add two Int objects and return new Int type object, here I am writing signature of this function for your convenience

     ```
     Int add (const Int & )
     ```

Write main and create two objects. One object using non-parameterized constructor and second using parameterized construct. Call add function and display the result of add function by calling display function.

2. Imagine a tollbooth at a bridge. Cars passing by the booth are expected to pay a 50 cent toll. Mostly they do, but sometimes a car goes by without paying. The tollbooth keeps track of the number of cars that have gone by (without paying), and of the total amount of money collected. Model this tollbooth with a class called TollBooth. The two data members are a type unsigned int to hold the total number of cars, and a type double to hold the total amount of money collected. A non-parameterized constructor initializes both of these to 0. A member function called payingCar() increments the car total and adds 0.50 to the cash total. Another function, called nopayCar(), increments the car total but adds nothing to the cash total. Finally, a member function called display() displays the two totals. Make appropriate member functions const. Write a main function to test this class. Run a loop and ask user (with appropriate messages) to press 1,2 or 0. For 1, call payingCar() function, for 2 call nonpayCar() function and for 0, terminate the loop. At the end display total cars and total cash and then exit.
3. Create a class called time that has separate unsigned integer data members for hours, minutes, and seconds. One non-parameterized constructor should initialize data members to 0. Second parameterized constructor should initialize data members by value of corresponding parameters by calling respective setter functions. Write setter functions for each data member to check range of values for 24-hour clock. For invalid values store zero. Write tow functions to display time, one in 24-hour format and other in 12-hour format (including AM & PM). Write main function to create two initialized time objects. Call display functions, then setter functions and again call display functions. Make appropriate member functions const.