ENGR:2730 Computers in Engineering
Bell & Christensen, Spring 2022, Homework #2

**Deadline: Friday, February 11, 2022 at 11:59pm**
**Early turn in bonus: Wed., February 9, 2022 at 11:59pm**

**Homework Goal:** Learn how to write and use simple C++ classes and how to unit test class methods.

To complete this homework, you need to modify the **Dog** class and main program provided in this assignment. It is recommended to work all parts of this homework together (i.e., test and document your code as it is written). It is also recommended that you commit your changes to SVN each time you write a method, test, and comments.

## Part 1. (20pts)

1. Implement a **getMonths** method following the Principle of Least Privilege.
2. Modify the **setMonths** method to validate the input parameter. If the input is less than zero, set the value of month to zero. Follow the Principle of Least Privilege when modifying this function.
3. Implement a **getHumanYears** method that returns the number of years as a double that the dog has been alive in human years.
4. Implement a **getDogYears** method that returns the number of years as a double that the dog has been alive in dog years. Assume that one human year is equal to seven dog years.
5. Define and implement a constructor for the Dog class. This constructor should have one input parameter with a default value of zero. The constructor should initialize the month member variable to the value of the input parameter unless it is negative. If the value of the input parameter is negative, the constructor should set the value of month to zero.

## Part 2. (10pts)

1. Document your Dog class and main function. You need to have a comment block at the top of your program with contains at a minimum: your name, the date, a description of the program, and descriptions of all inputs and outputs.

## Part 3. (20pts)

Rewrite the given main program to unit test your Dog class. The main program should hard code the creation of multiple Dog objects and calls to the object methods (see example below). Please ensure you code enough test cases to make sure all your functions work.

1. Your main function must instantiate at least four (4) Dog objects and fully test that the constructor work properly. Your main function must test each function (getMonths, setMonths, getHumanYears, getDogYears) of the Dog class with no fewer than four (4) tests each. *Note: You do not need 4 tests of each function for each of the 4 objects. You simply need to test each function 4 times across any of the 4 objects. Please group similar unit test together, i.e., place all constructor tests together, etc.*
2. Each unit test should print the expected output followed by the actual output. Examples:

```
Dog bess;
cout << "Expected bess.getMonths() to return 0. Actual value is " << bess.getMonths() << endl;

Dog truman(14);
cout << "Expected truman.getMonths() to return 14. Actual value is " << truman.getMonths() << endl;

truman.addBirthday();
cout << "Expected truman.getStage() to return Adulthood. Actual value is " << truman.getStage() << endl;
```

The code below is adapted from zyBooks C++ example 8.5.1: Mutators, accessors and private helpers. Modify this code to do your homework. Type this code into CLion instead of copying and pasting into CLion to avoid problems.

```
#include <iostream>
using namespace std;

class Dog {
public:
   void setMonths(int monthsToSet);
   void addBirthday();
   string getStage() const;
private:
   int months;
};

void Dog::setMonths(int monthsToSet) {
   months = monthsToSet;
}

void Dog::addBirthday() {
   months += 12;
}

string Dog::getStage() const {
   string stage;
   if (months < 11) {
      stage = "Puppy";
   }
   else if (months < 25) {
      stage = "Adolescence";
   }
   else if (months < 80) {
      stage = "Adulthood";
   }
   else {
      stage = "Senior";
   }

   return stage;
}

int main() {
   Dog buddy;

   buddy.setMonths(54);
   cout << buddy.getStage() << endl;

   buddy.addBirthday();
   buddy.addBirthday();
   buddy.addBirthday();

   cout << buddy.getStage() << endl;
   return 0;
}
```

**Submission Instructions**

- Make a CLion project called "hw2" (ALL LOWERCASE) in your SVN homework directory.
- Check your homework into SVN.

*Hint: you can see the current version of your submission by opening this link in a web browser:*
*https://class-svn.engineering.uiowa.edu/cie/projects/spring2022/*