ITP 165: Introduction to C++ Programming

Homework 10: Animal Training — Due: 11:59pm, November 13th, 2015

Goal

For this assignment, you will temporarily say goodbye to your unique animals and you will, instead, create a new generic Animal class from the given Animal.h header file. You will also be gaining practice using dynamic classes in your main() function.

Setup

- Create a new project in Visual Studio or Xcode called **homework10** and add three new files named **trainer.cpp, Animal.h,** and **Animal.cpp**. The **Animal.h** header file has already been given to you with your assignment file.
- Each file must begin with the following (replace the name and email with your actual information):

```
// Name
// ITP 165, Fall 2015
// Homework 10
// USC email
```

- Use the input file "Animal.txt" or "Animal_Mac.txt" (PC or Mac, respectively) included with this assignment file as sample input to your program.

Part 1: Animal class – Animal.cpp

- Be sure to read the `Animal.h` file to understand the class you are implementing. It will be your job to create the corresponding Animal.cpp file, which has:
  - 3 constructors: `Animal.h` outlines what each of these does
  - Getters for each private member variable
  - Setters for each private member variable: `Animal.h` outlines the allowed ranges for each variable – you will need to put limits on the values allowed in your setters
  - A Load function: look in `Animal.h` for function implementation comments
  - A Save function: look in `Animal.h` for function implementation comments

Part 2: Animal training – trainer.cpp

- For your third file, you need to be sure to include all libraries needed. For this assignment you will need to include `"Animal.h"`, `<iostream>, <cstdlib>,` and `<ctime>` as we will be using random numbers.
- First you will make the `PrintStats` function outside of your class and before main()
  - This function will return nothing and accepts an Animal pointer as a parameter
  - Its purpose is to display all of the member variables of the animal
- Next will be your standard int main() function. Please seed your random number generator with the standard srand() function.
- Declare an Animal pointer and initialize it to a null pointer.

- Now you will display a menu of options for your user, this time with 7 options:
    - Save and Quit
        - This case asks the user for an output file name, then calls the corresponding `Save` function on the animal pointer. Tell the user when the save is complete and make sure to clean up your memory before quitting.
    - New Animal
        - This case will tell the user a new animal is being created, then will dynamically allocate memory on the heap with the Animal class's default constructor. Print the stats of the newly created animal for your user.
    - Load Animal
        - This case will ask the user for an input file name, then will dynamically allocate memory on the heap with one of the Animal class's non-default constructors. Print the stats of the newly created animal for your user.
    - Rename Animal
        - This case will ask the user for a new name, then will set your animal pointer's name with this new value. Print the stats of the newly named critter.
    - Print Stats
        - This case will call your `PrintStats` function that you created above main with your animal pointer passed in as a parameter.
    - Heal Animal
        - This case will increase the Animal's current health with a random number between 1 and 10. Tell the user how much the animal was healed by and also print the stats of the animal after healing.
    - Train Animal
        - This case will not be implemented – tell your user it is coming soon!
    - Be sure to have a default case to catch invalid input
- Loop over the menu options until the user has entered the Save and Quit option.

Full Sample Output

Below is sample output for a full run-through of the program. Your output should resemble the following – but doesn't need to match it exactly (user input is in **red**).

```
Welcome to ITP 165's Training Arena!

        MENU OPTIONS:
0. Save and Quit
1. New Animal
2. Load Animal
3. Rename Animal
4. Print Stats
5. Heal Animal
6. Train Animal: COMING SOON
What would you like to do? 1
```

```
Creating new animal...
Welcome your new animal:
******************
Name: Animal
Health: 70
Power: 60
Armour: 50
******************

        MENU OPTIONS:
0. Save and Quit
1. New Animal
2. Load Animal
3. Rename Animal
4. Print Stats
5. Heal Animal
6. Train Animal: COMING SOON
What would you like to do? 2

File input name? Animal.txt
Animal loaded!
******************
Name: Generic Name
Health: 50
Power: 75
Armour: 50
******************

        MENU OPTIONS:
0. Save and Quit
1. New Animal
2. Load Animal
3. Rename Animal
4. Print Stats
5. Heal Animal
6. Train Animal: COMING SOON
What would you like to do? 3

New name: My Awesomest Animal
New name set.
******************
Name: My Awesomest Animal
Health: 50
Power: 75
Armour: 50
******************

        MENU OPTIONS:
0. Save and Quit
1. New Animal
2. Load Animal
3. Rename Animal
4. Print Stats
5. Heal Animal
6. Train Animal: COMING SOON
What would you like to do? 4
```

```
******************
Name: My Awesomest Animal
Health: 50
Power: 75
Armour: 50
******************

        MENU OPTIONS:
0. Save and Quit
1. New Animal
2. Load Animal
3. Rename Animal
4. Print Stats
5. Heal Animal
6. Train Animal: COMING SOON
What would you like to do? 5

Let's see what your healing roll is...
Healed by 1!
******************
Name: My Awesomest Animal
Health: 51
Power: 75
Armour: 50
******************

        MENU OPTIONS:
0. Save and Quit
1. New Animal
2. Load Animal
3. Rename Animal
4. Print Stats
5. Heal Animal
6. Train Animal: COMING SOON
What would you like to do? 6

Feature coming soon!

        MENU OPTIONS:
0. Save and Quit
1. New Animal
2. Load Animal
3. Rename Animal
4. Print Stats
5. Heal Animal
6. Train Animal: COMING SOON
What would you like to do? 7

Invalid choice. Try again.

        MENU OPTIONS:
0. Save and Quit
1. New Animal
2. Load Animal
3. Rename Animal
4. Print Stats
5. Heal Animal
6. Train Animal: COMING SOON
What would you like to do? 0

File output name? MyAnimal.txt
```

```
Save complete! Time to quit.
Thanks for playing! Goodbye.
Press any key to continue . . .
```

A Note on Style

Be sure to comment your code. Also, remember to comment out any extra debug statements.

As we discussed in lecture, it is extremely important that your code is properly indented as it greatly adds to readability. Because of this, if you submit a code file that is not reasonably indented, you will have points deducted.

Likewise, you will lose points if your variable names are not meaningful. Make sure you use variable names that correspond to what you are actually storing in the variables.

Deliverables

1. A compressed folder containing only the **trainer.cpp, Animal.h,** and **Animal.cpp** files, named **HW10**. This can be done by:
    a. WINDOWS:
        i. Select all your files
        ii. Right click
        iii. Send to ->
        iv. Compressed (zipped) folder
        v. Rename this zipped folder to HW10
        vi. Submit this zipped folder through Blackboard
    b. OSX:
        i. Select all your files
        ii. Right click
        iii. Compress 3 items
        iv. Rename this Archive.zip to HW10
        v. Submit this zipped folder through Blackboard

Grading

Item	Points
Part 1: Animal class – Animal.cpp	20
Part 2: Animal interaction – trainer.cpp	20
Total*	**40**

* Points will be deducted for poor code style, or improper submission.