CS 201: Problem Solving & Programming II
Lab #4

The warm up task

In this lab, we will go through the Class Inheritance one more time. A very common example of inheritance is classification of animals into different categories. We will use this scenario to categorize the animals into different classes. Let's think, *Animal* is a class which holds the very common attributes that all types of animal share, for example, weight and length. Then we will create more specific classes to define different types of animals.

The Animal Class

1. First create an Animal class using the following diagram. Recall from our last lab that, the private members are denoted with – sign, public members like public methods are denoted with + sign. An UML diagram of the Animal class is given below:

Animal
-m_weight: double -m_length: double
+Animal() +Setup(weight: double, length: double): void

2. Create two files, Animal.h and Animal.cpp. The header file (Animal.h) will contain only the structure and declaration of the class.

Note: A sample declaration of a class with method:

```
class Animal {
public:
   void setWeight(double weight);
Private:
   double m_Weight;
};
```

3. Write the actual method definitions of the class in the Animal.cpp file. Don't forget to add the Animal.h in the #include section.

Note: A sample definition of the methods of Animal class can be:

```
void  Animal::setWeight(double weight) {
   Animal::m_Weight = weight;
}
```

4. When you are done creating the two classes, open a new file **main.cpp** and write a **main()** function.
5. Inside the *main* function create an object of the Animal class and call the setup function to pass weight and length of the animal.
6. Compile and run the program. If you are getting error, try to fix them. Take help from the instructor if necessary.

The Exercise

Suppose you are an in-charge of the museum, and the manager asked you make a list of all the specimens of the gallery. Fortunately, right now only three types of artifacts are there – Humans, Birds and Dogs.

Now you have decided to create a class for each type and store the attributes as member variables of the class. You have noticed that there are some common attributes for all types. So you decided to create a common class Animal for that.
Therefore, **<u>you need to create four classes: Animal, Human, Bird, Dog</u>**. We already have the Animal class defined. So, we have to define the remaining three classes. The structures of the classes are given as UML diagram below:

Human
-m_ethnic: string -m_language: string
+Human() +Setup(ethnic: const string&, language const string&, weight: double, length: double): void +getEthnic(): string +getLanguage(): string

Bird
-m_name: string -m_canFly: bool -m_foodHabit: string
+Bird() +Setup(name:string, canFly: bool, foodHabit: const string&, weight: double, length: double) : void +getCanFly(): bool +getFoodHabit() : string

Dog
-m_breed: string -m_lifeSpan: double
+Dog() +Setup(breed: const string&, lifeSpan: double, weight: double, length: double): void +getBreed(): string +getSize(): string

Creating the three classes:

1. Create the three classes according to the UML diagram structure shown above.
2. For each of the class, create two files- header and cpp. For example, Human.h, Human.cpp etc.
3. **All three classes (Human, Bird and Dog) should inherit from the Animal class.** Recall from the previous lab, how to declare a class to inherit from another class. Also note that the base class should be defined before you can inherit it. Use **#include** statement to add the base class you want to inherit from.

```
#include “Animal.h”
class Human: public Animal {
   // declaration of the Human class goes here...
};
```

4. When you are done creating all three classes (six files in total excluding Animal class), compile and run the program. Again, if error pops up, try to fix them.

   If you are done creating all the classes move to the next section.

Taking input from the file:

By now, you should have four classes defined properly. Now move to main.cpp file. We already have some code written here in the main() method. You may remove them or keep them within a comment block.

Now, you have to define three methods like the following and call them from main() method:

```
LoadHumans();

LoadBirds();

LoadDogs();
```

These methods will take input from three files and create objects accordingly. For example, LoadHumans() method will read the "humans.txt" file and create objects of Human class. Similarly, the LoadBirds() and LoadDogs() will create objects of Birds and Dogs respectively.

Creating Human objects:

The content format of the human file (humans.txt) is:

```
Ethnic
Language
Length
Weight
```

There will be 3 entries of human in the file, so you have to create three objects of Human type. You may define an array of length three for it.

Creating Bird objects:

The content format of the bird file (birds.txt) is:

```
name
Can fly
Life span
length
Weight
```

There will be 3 entries of bird, so you have to create three objects of Bird type. Note that, "Can fly" values will be 0 or 1. You can check this value and set the "canFly" variable to true/false accordingly.

Creating Dog objects:

The content format of the dog file (dogs.txt) is:

```
breed
lifespan
length
Weight
```

There will be 3 entries of dog, so you have to create three objects of Dog type.

Well done! The exercise is complete. Make sure your code compiles without error and creates the objects properly. There's an extra credit section in the next page if you want to boost up your score.

Extra credit

Display the objects in following manner:

```
The little Museum
Specimen list:
* Humans (3)
* Birds (3)
* Dogs (3)
```