CS 53 – Introduction to Programming
Homework Assignment #8

Instructions:

(1) This assignment is **due by noon on Monday, December 13, 2004**. Late assignments will not be accepted.

(2) You must submit your solution through the *Digital Drop Box* on the Blackboard web site (blackboard.umr.edu). Directions for doing this are given at the end of this document.

(3) This assignment will be worth 5% of your course grade.

For this assignment you are to create a superclass (or base class) called *GameEntity* and a subclass (or derived class) called *Monster*. Specifically, you are to do the following:

1. Create a file called GameEntity.h
2. Define in GameEntity.h a class called *GameEntity* with **private** member variables *game* (a string) and *name* (a string).
3. In the *GameEntity* class, use enum to define two **public** constants: one for MAX_NAME_LENGTH = 25 and one for MAX_GAME_NAME_LENGTH = 10.
4. In the *GameEntity* class, define two **public** constructors: (1) a default constructor that takes no parameters and initializes the *name* to "(No Name)" and *game* to "(No Game)", and (2) a constructor that takes 2 parameters *theGame* and *theName*, and sets the corresponding member variables *game* and *name* to those values.
5. In the *GameEntity* class, define **public** accessor and mutator functions for each of the member variables.
6. In the *GameEntity* class, define a **public** member function to overload the == operator. This function should return true if the *name*s are the same and the *game*s are the same; otherwise, it should return false.
7. In the *GameEntity* class, define a **public** member function to overload the extraction >> operator. It should prompt the user to enter values for the *game* and *name*.
8. In the *GameEntity* class, define a **public** member function to overload the insertion << operator (to output the *game* and the *name*).
9. Create a file called ***Monster.h*** and in it define a class *Monster*. Make this class be a **public** subclass of the class *GameEntity*.
10. In the *Monster* class define **private** member variables for *weapons* (a string) and *healthRating* (an integer).
11. In the *Monster* class, use enum to define a **public** constant for MAX_WEAPONS_LENGTH = 25.
12. In the *Monster* class, define **public** accessor and mutator functions for each of the member variables.

13. In the *Monster* class, define a **public** default constructor that takes no parameters, and initializes *weapons* to "(No Weapons)" and *healthRating* to zero.
14. In the *Monster* class, define a **public** constructor *Monster(const string theName, const string theGame, const string theWeapons, const int theRating)*. We want this constructor to initialize the appropriate member variables from both the *GameEntity* and *Monster* classes to the corresponding parameters. In order to get the *Monster* constructor to also call the appropriate *GameEntity* constructor, the definition of this constructor should look like this:

```
Monster::Monster(const string theName, const string theGame,
                 const string theWeapons, const int theRating) :
        GameEntity(theName, theGame)
{
weapons = theWeapons;
healthRating = theRating;
}
```

15. In the *Monster* class, define a friend function ***friend istream& operator >>(istream& ins, Monster& m)*** to overload the extraction operator. This function should prompt the user to enter the *name*, *game*, *weapons*, and *healthRating* for a monster. **Call the extraction operator for *GameEntity* to get the input for name and game.**
16. In the *Monster* class, define a friend function ***friend ostream& operator <<(ostream& outs, Monster& m)*** to overload the insertion operator. This function should neatly output the *name*, *game*, *weapons*, and *healthRating* for a monster. **Call the insertion operator for *GameEntity* to output the name and game.**
17. Remember to add ***#include "GameEntity.h"*** to the ***Monster.h*** file so that it will be able to use the definitions from ***GameEntity.h***

We will test your class definitions by running our program hw8.cpp (which is posted on the Blackboard web site for HW #8). You are not allowed to modify that file!

Directions for Submitting Your Assignment for Grading

For this assignment, you are to submit your program using the *Digital Drop Box* on the Blackboard web site. This is **NOT** the method that you use in CS 54 to submit your lab projects for grading!

To submit your program for grading do the following:

(1) Compile and run your program using the GNU (g++) compiler.

(2) Thoroughly test your program on your own.
(3) Login to the CS 53 Blackboard web site (blackboard.umr.edu).
(4) Select *Tools*.
(5) Select *Digital Drop Box*.
(6) Click on *Send File*.
(7) For *Title:* enter **CS 53 – HW #8**
(8) For *File:* use the *Browse* dialog to select your ***GameEntity.h*** file from your computer.
(9) Click on *Submit*.
(10) On the next page that appears (which should say *Receipt: Success*), click on *OK*.
(11) Repeat steps 4-10 for your ***Monster.h*** file.

If you have problems, contact your instructor (leopoldj@umr.edu) or the TA (Rong Zhuge, rzwr6@umr.edu).

You may submit your program any time before the due date, but please only submit it once. Do **NOT** wait until the last minute to submit your work in case you encounter problems!