Assignment 3 – Inheritance

Goals-

Identify requirements for a program using polymorphism
Create a program to demonstrate your class hierarchy

You will create a simple class hierarchy as the basis for a fantasy combat game. Your 'universe' contains Goblins, Barbarians, Reptile People, Blue Men and others. Each will have characteristics for attack, defense, armor, and strength points.

Type	Attack	Defense	Armor	Strength Points
Goblin[4]	2d6 *Achilles	1d6	3	8
Barbarian[1]	2d6	2d6	0	12
Reptile People[2]	3d6	1d6	7	18
Blue Men[3]	2d10	3d6	3	12
The Shadow[3]	2d10	1d6	*Special	12

3d6 is rolling three 6-sided dice. 2d10 is rolling two 10-sided dice.

*Special- The Shadow can manipulate perceptions of others. For a given attack there is a 50% chance that The Shadow is someplace else and no damage is received. (If you want to be more consistent with the original radio plays or later comics it would be 90% or higher. In that case there would be no defense roll as he had flaring overcoat but little else to protect him.) Note- Implement this AFTER you have tested the base defense function and save a copy of that file. Just in case. ☺

*Achilles- When fighting anything but another Goblin they might get lucky and cut an Achilles Tendon. If a Goblin rolls a 12 in attack then the target has his/her attack roll halved for the remainder of the combat. Blue Men attack in groups so this means one of them are knocked out. Note- Implement this AFTER you have tested the base attack function and save a copy of that file. Just in case. ☺

To resolve an attack you will need to generate 2 dice rolls. The attacker rolls the appropriate number and type of dice under Attack. The defender rolls the appropriate number and type of dice under Defense. You subtract the Defense roll from the Attack roll. That is the damage.

To apply the damage you subtract the Armor value. The result is then subtracted from the Strength Points. That value becomes the new Strength Points for the next round. If Strength Points goes to 0 or less then the character is out of the combat. Remember that for The Shadow there may be 0 points inflicted based on his special ability.

You need to create a Creature class. Then you will have a subclass for each of these characters. Note that the Creature class will be an abstract class. You will never instantiate one. For our purposes right now each subclass will vary only in the values in the table. Since each starts with the same data elements you will only need one constructor. It is part of your design task to determine what functions you will need. The only value that can change is the Strength Points.

This is the first stage in what will be a larger project. Please do not add any creatures of your own.

You must complete your design document. In that document and in your reflections you can discuss how the original design may have changed as you worked through the problem. You must also submit a test plan. The test plan should cover all logic paths. So you should have each character type have combat with all character types (including another of its own). Remember to submit these documents as PDF files.

It is not hard, just a lot to think about. The TAs will be asked to grade your project against your design so please do not just throw together some random stuff so you have a file to submit. No, you are not required to implement only the design you submit. BUT, your reflections will need to explain the difference. So the old adage garbage in, garbage out will not apply here. If you give us a random design you will need to explain

each step in how you got to the code submitted. In other words, that will make it much more difficult. So, learn a good habit and think about it before you start coding. ☺

HINT: This program has a random element. You will need to address that in your test plan. It will also affect debugging. Your design should address this (potential) problem. It is not hard but you need to think about it.

What you need to submit:

Your program file(s) with the implementation of these five creatures inheriting from a single parent.
Your design document (including the class hierarchy)*
Your test plan*
Your reflections document- including the design and test documentation

1. Big, powerful, with tough skin. But very slow.

2. Think Conan or Hercules from the movies. Big sword, big muscles, bare torso.

3. They are small (6" tall), fast and tough. So they are hard to hit and can take some damage. As for the attack value, you can do a LOT of damage when can crawl inside the armor or clothing of your opponent. ☺ And yes they are the Nac Mac Feegle of Discworld. I just wanted a shorter name for your project.

4. They are small, not fast, and not strong.

NOTE: **The sample creatures are unbalanced intentionally**. This will help you in debugging your program! Some will win a lot, and others will lose a lot.

======

Grading:

- programming style and documentation (10%)
- create the base class and the Barbarian class (25%)
- create the Goblin class- overload or redefine attack function (10%)
- create the Reptile People class (5%)
- create the Blue Men class (5%)
- create the Shadow class- overload or redefine defense function (10%)
- create a test driver program to create character objects which make attack and defense rolls- required to show your classes work correctly (15%)
- reflections document to include the design description, test plan, test results, and comments about how you resolved problems during the assignment (20%)

Suggestion: The grading is set up to encourage you to develop your program incrementally! Start with the base and Barbarian classes. Create the testing program that runs sample combats for you toe test your code. How do you handle random die rolls when testing your code? Then do Reptiles and Blue Men. Then do Goblins and Shadow, without their special abilities. Then add the special attack and defense features using polymorphism.

Hint: Create your design before coding anything! You should even be outlining your test plan. At each step of the process make notes about what worked, what changed. Doing this as you progress will make writing the reflections easier.