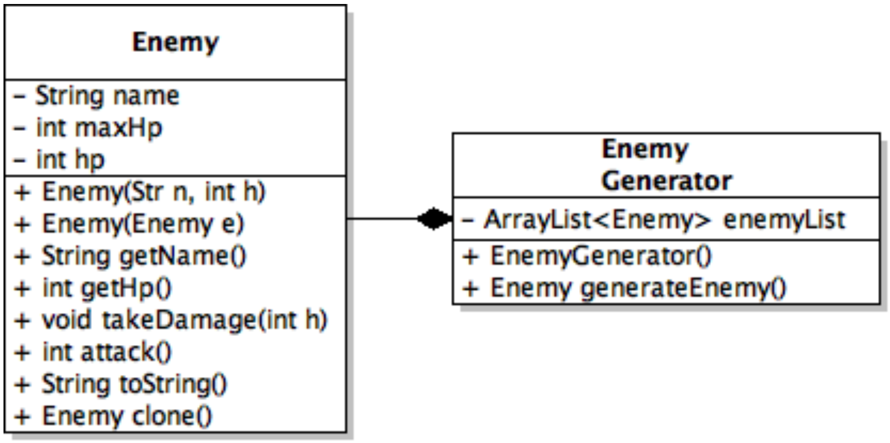
# CECS 277 – Lab 12 – Prototype

## Monster Fighter

Create a program that generates new Enemies for the user to fight. The user will attack the enemy, and if the enemy is still alive, it will attack the user back. Keep track of the user's hit points (hp), if the user's hit points reaches 0, then the game ends. Keep a count of the number of enemies the user has slain and report that total when the game ends.

Implement the program using the following UML diagram:



**Classes**:

1. Enemy:
    a. Has instance variables for the name and the hp.
    b. Has two constructors:
        i. one passes in name and hp, initialize the hp and maxHp to the same starting value.
        ii. the other one is a copy constructor for the prototype.
    c. Has two get methods for name and hp.
    d. Method takeDamage passes in an amount of damage to subtract from the hp. The hp should not go below 0, if it does, set it back to 0.
    e. Method attack() – returns a random amount of damage that the enemy does. You can choose the range (ex. 1-5).
    f. Method toString() – return the name and hp (ex. Orc 2/5 hp)
    g. Method clone() – creates and returns a copy of the enemy by calling the copy constructor.
2. EnemyGenerator:
    a. Has an ArrayList of template enemies.
    b. Constructor reads in a file (enemyList.txt) of template enemies and stores them in the ArrayList.
        i. Format of file is 'name,maxHp' and has one enemy per line.
    c. Method generateEnemy – chooses a random enemy from the ArrayList and creates a copy of it to return.
3. Main:
    a. Create an EnemyGenerator object and integers for the user's hp and counter to keep track of the number of enemies slain. You can start the

    user’s hp at any value you like (ex. 25).

  b. Create an Enemy object and prompt the user to attack (you can include multiple types of attacks that each do a different amount of random damage for the user to choose from). Do a random amount of damage to the enemy, and if the enemy’s hp is greater than 0, then it can attack back. Repeat until the enemy’s hp is 0. Once it is 0, then increment the counter and generate the next enemy to attack.
  c. Repeat until the user quits or runs out of hp.

**Notes**:

1. The enemy should not be able to attack back if it doesn’t have any hp.
2. The user should not be able to attack the enemy if the user doesn’t have any hp.
3. Only generate a new enemy after the previous one is slain (ie. you’ll need an additional loop for this).
4. You should not return an Enemy directly from the ArrayList. You should always make a clone of it first.
    a. The reason for this is that if you return one of the template enemies, and the user does damage to it, when that enemy is chosen again later, then it will still have 0 hp, rather than the full hp it should start with. Calling clone will make sure that the template enemies are never touched (test this out by not calling clone and seeing that respawned enemies have 0 hp).
5. Error check the user’s input for the menu.

**Example Output**:

```
You have 25/25 hp.
You encounter a Orc
HP: 4/4
What do you want to do?
1. Attack Enemy
2. Quit
1
You attack Orc for 3 damage.
Orc attacks you for 3 damage.
Orc
HP: 1/4
What do you want to do?
1. Attack Enemy
2. Quit
1
You attack Orc for 4 damage.
You have slain the Orc

You have 22/25 hp.
You encounter a Goblin
HP: 2/2
What do you want to do?
1. Attack Enemy
2. Quit
1
You attack Goblin for 5
damage.
You have slain the Goblin

...

You have 1/25 hp.
You encounter a Froglok
HP: 2/2
What do you want to do?
1. Attack Enemy
2. Quit
1
You attack Froglok for 1
damage.
Froglok attacks you for 1
damage.
You have died.
Enemies Slain: 23
Game Over.
```