

Program Description: Battle!

In this project you will be adding to your **Player/Enemy** classes from your previous assignment, and creating a simple GUI using swing components to display your output. You will first need to add an **attack() method** which will use your Player's stats to damage the Enemy. The attack() method should take in an Enemy object (passed as a Player for polymorphic intent) as a parameter to cause damage to it directly. An example of the method could follow a structure similar to this:

```
public void attack(Player enemy)
{
    enemy.setHealth( enemy.getHealth( ) - this.getDamage( ));
}
```

NOTE- You must use get and set methods to effectively access and change values passed by the Enemy class. The above code will set an enemy's new health by subtracting the player's damage attribute from the enemy's current health value. You may also include the enemy's defense value in the calculation to make things more interesting.

Once the attack() method is created, you will need to create a basic **JFrame** for display and use swing components to output a created Player's stats and a created Enemy's stats. The JFrame will include all graphical components as well as the creation of both your Player and Enemy objects. You will still require a **Driver** class to instantiate the initial JFrame object.

To handle the player/enemy creation, you may either create a second JFrame to be called first, or you may simply utilize JOptionPanels to request the needed information for the player before the stat frame is display.

You will also be required to add a **button labeled "Attack"** that will call your attack() method, damage the Enemy, and re-display the stats for both the Player and Enemy. After your player has attacked, have the Enemy attack back and damage the player. This process will repeat for each press of the "Attack" button until either the Player or Enemy has died (health<=0)!

Example Output:**Bonus: JApplet Modification +10pts**

You will adapt your code to run as a JApplet. You will need to create an `init()` method that will create your initial window to display your content. You will also need to create an HTML file to run your newly created JApplet.

All other base functionality will remain the same.

Constraints:

- a) Make sure you use the following comment at the beginning of your program:

```

/*****
Homework 7:   Battle!
Program By:   YOUR NAME
File Name:    Player.java or Enemy.java or Driver.java
Function:     This program creates player and enemy combatants and has them fight!
*****/

```

- b) Code should be separated into **THREE (3)** separate files, One for each class
- c) **ATTACH** your completed .java files to the *submission link*. Do not cut and paste your code into the message section.