

Individual Programming 4

Start Assignment

Due Saturday by 1pm **Points** 20 **Submitting** a file upload **File Types** py
Available Nov 13 at 1:30pm - Nov 27 at 1:30pm 14 days

This is an individual programming assignment with four main parts. Although you are welcome to collaborate and exchange ideas with others, all work submitted for this assignment should be your own.

The following four-part task is worth 20 points, no extension required. **START EARLY! THIS IS AN EXTENSIVE PROGRAM.**

Note, each part does not have the same grading weight. In addition, names and capitalization of the classes and functions are graded for accuracy (imagine that another programmer is working with your code too--these are the naming conventions that you two agreed on).

Be sure to watch the lecture as there are hints (and code that you can use) for this assignment.

Part 1 (Super Class):

Write a Character class that represents characters in a game. The character should have a name, which is a required argument. It should also have level (lvl), health points (hp), attack level (at), defense level (df), and experience (exp). These values are all optionally initialized by the user. By default, level is 1, health points is 100, attack level is 10, defense level is 1. Keep in mind that you may need to initiate extra variables (such as the max HP).

The Character class should have the following functions:

```
rename(), renames the character
setlvl()
sethp()
setat()
setdf()
setexp()
getname()
getlvl()
gethp()
getat()
getdf()
getexp()
attack(), details below
```

The attack() method should take in one argument (another Character), whose hp you will be depleting by the attacking Character's attack points minus the defending Character's defense points. That is if Alpha (below) attacks Beta, Beta will lose 8 HP (10 Alpha_attack - 2 Beta_defense). At the end of each attack, print out who attacked who and current HP stats of the two characters. Note, HP cannot be more than the max HP or less than 0. In the attack method, you must also print "__ defeated __" if any one character's HP reaches 0.

Part 2 (Inheritance 1):

Write a MajorCharacter subclass that inherits the Character class. The MajorCharacter subclass keeps all methods of the Character class, but it has four additional methods. Both recover() and fullheal() print out the recovery status (see example further below):

```
recover(), takes in one integer argument and recovers the current character's HP by that amount (up to the max HP)
fullheal(), recovers the current character's HP to the max HP
shield(), protects the character from any damage for three attacks (you'll need to create a turn counter variable and modify the MajorCharacter's
sethp() method -- not the Character's sethp() method -- to account for how many times the character has been attacked). Shields are not stackable
(though it's up to you whether you want to reset it to 3 when used again or continue from the current status), so characters can only protect them
selves for a max of the next three attacks.
```

```
charge(), multiply's the character's next attack by 2.5 (you'll need to create a boolean variable to indicate whether the character is charged or not). Note, charges are not stackable. That is, if a character's already charged, charging again will do nothing.
```

Note, you cannot have a variable and a function in a class with the same name (e.g. don't use a variable named `self.shield` because you'll already have a function called `shield()`).

To test your code, consider the following two characters:

```
Alpha
Level: 1
HP: 100
Attack: 10
Defense: 5
Exp: 100
```

```
Beta
Level: 1
HP: 106
Attack: 11
Defense: 2
Exp: 100
```

Below, you are given some test code. **Do not include the following test code in your submission.** This is provided to check your work along the way. Your classes should work so that the following lines of code run:

```
mainchar = MajorCharacter("Alpha", df = 5, exp = 100)
opponent = MajorCharacter("Beta", hp = 106, at = 11, df = 2, exp = 100)
while (mainchar.gethp() > 0 and opponent.gethp() > 0):
    mainchar.attack(opponent)
    if opponent.gethp() > 0:
        opponent.attack(mainchar)

mainchar.recover(40)
mainchar.recover(40) ## note that mainchar only recovers to 100/100 (the max)
opponent.fullheal()
```

The code above should produce the following printout:

```
Alpha attacked Beta. Alpha's HP is 100/100. Beta's HP is 98/106.
Beta attacked Alpha. Beta's HP is 98/106. Alpha's HP is 94/100.
Alpha attacked Beta. Alpha's HP is 94/100. Beta's HP is 90/106.
Beta attacked Alpha. Beta's HP is 90/106. Alpha's HP is 88/100.
Alpha attacked Beta. Alpha's HP is 88/100. Beta's HP is 82/106.
Beta attacked Alpha. Beta's HP is 82/106. Alpha's HP is 82/100.
Alpha attacked Beta. Alpha's HP is 82/100. Beta's HP is 74/106.
Beta attacked Alpha. Beta's HP is 74/106. Alpha's HP is 76/100.
Alpha attacked Beta. Alpha's HP is 76/100. Beta's HP is 66/106.
Beta attacked Alpha. Beta's HP is 66/106. Alpha's HP is 70/100.
Alpha attacked Beta. Alpha's HP is 70/100. Beta's HP is 58/106.
Beta attacked Alpha. Beta's HP is 58/106. Alpha's HP is 64/100.
Alpha attacked Beta. Alpha's HP is 64/100. Beta's HP is 50/106.
Beta attacked Alpha. Beta's HP is 50/106. Alpha's HP is 58/100.
Alpha attacked Beta. Alpha's HP is 58/100. Beta's HP is 42/106.
Beta attacked Alpha. Beta's HP is 42/106. Alpha's HP is 52/100.
Alpha attacked Beta. Alpha's HP is 52/100. Beta's HP is 34/106.
Beta attacked Alpha. Beta's HP is 34/106. Alpha's HP is 46/100.
Alpha attacked Beta. Alpha's HP is 46/100. Beta's HP is 26/106.
Beta attacked Alpha. Beta's HP is 26/106. Alpha's HP is 40/100.
Alpha attacked Beta. Alpha's HP is 40/100. Beta's HP is 18/106.
Beta attacked Alpha. Beta's HP is 18/106. Alpha's HP is 34/100.
Alpha attacked Beta. Alpha's HP is 34/100. Beta's HP is 10/106.
Beta attacked Alpha. Beta's HP is 10/106. Alpha's HP is 28/100.
Alpha attacked Beta. Alpha's HP is 28/100. Beta's HP is 2/106.
Beta attacked Alpha. Beta's HP is 2/106. Alpha's HP is 22/100.
Alpha attacked Beta. Alpha's HP is 22/100. Beta's HP is 0/106.
Alpha defeated Beta!
Alpha's HP has been restored to 62/100.
Alpha's HP has been restored to 100/100.
Beta's HP has been fully restored (106/106).
```

Part 3 (Inheritance 2):

* Add a Minion class which inherits all of the features of the Character class.

* Add a Boss class which inherits all of the features of the Character class. Add the following method to the Boss class:

```
spawn(), generates a Minion named "Minion" with HP and attack equal to 1/4 (rounded down) of the Boss's max. The Minion should be the same level as the Boss but have 0 exp and 0 defense.
```

Part 4 (Game Program):

Game Program:

If you haven't already, be sure to add a print statement in your each of the action methods (attack, recover, fullheal, shield, charge, and spawn) that lets players track the status of the game.

The program should run in the following order:

1. Start your program by asking the user how many players he/she would like to create. Only allow 1-8 players (note, this must be an integer). Ask the user for all player names. Create each player with lvl = 1, hp = 100, at = 10, df = 5, exp = 0. Store all players in a list.
2. Create a Boss (named "Boss") that has HP equal to 20 times the number of players. If 3 players were created in part 1, then the Boss should have HP equal to 60. The level should be 1, attack should be 100, defense should be 0, and exp should be 0.
3. The Boss first spawns a Minion. Store all Minions in a list.
4. The program will loop and ask for each (alive) player's move. Use a list to keep track of alive players. Players can choose which Minion to attack. Use a list to keep track of the alive Minions. Players cannot attack the Boss until all Minions are gone.
5. All Minions (if any) will attack the player with the lowest HP. If there are multiple players with the lowest HP, the Minions will attack the latest created player (so Player 8 if there are 8 players with the same HP).
6. Repeat Steps 3 through 6 until the boss has been defeated or all players have been defeated.

```
## There's a few different ways to complete the tasks outlined, but here's a few hints if you're stuck
```

```
## Optional Hint 1: You can store a list of players as follows
```

```
playerlist = []  
newplayer = Player(__, __, __)  
playerlist.append(newplayer)
```

```
## Optional Hint 2: You can spawn the Minion in the Boss class and return it
```

```
def spawn():  
    newminion = Minion(__, __, __)  
    print("Boss spawned Minion.")  
    return newminion
```

Below you'll see three example outputs, with user-input in red. Note in Example 3, the user might put in invalid inputs. Please prompt the user again for a valid input.

Example 1:

```
How many players? 2  
Player 1 Name: Python  
Player 2 Name: Java  
Boss spawned Minion (HP 10, AT 25, DF 0)  
1. Attack  
2. Charge  
3. Shield  
Python, what would you like to do? 1  
1. Minion (HP 10)  
Python, who would you like to attack? 1  
Python attacked Minion. Python's HP is 100/100. Minion's HP is 0/10.  
Minion has been defeated.  
1. Attack  
2. Charge  
3. Shield  
Java, what would you like to do? 3  
Java is safe from the next 3 attacks.  
There are no Minions to attack.  
Boss spawned Minion (HP 10, AT 25, DF 0)  
1. Attack  
2. Charge  
3. Shield  
Python, what would you like to do? 1  
1. Minion (HP 10)  
Python, who would you like to attack? 1  
Python attacked Minion. Python's HP is 100/100. Minion's HP is 0/10.
```

```

Minion has been defeated.
1. Attack
2. Charge
3. Shield
Java, what would you like to do? 2
Java charged.
There are no Minions to attack.
Boss spawned Minion (HP 10, AT 25, DF 0)
1. Attack
2. Charge
3. Shield
Python, what would you like to do? 1
1. Minion (HP 10)
Python, who would you like to attack? 1
Python attacked Minion. Python's HP is 100/100. Minion's HP is 0/10.
Minion has been defeated.
1. Attack
2. Charge
3. Shield
Java, what would you like to do? 1
1. Boss (HP 40)
Java attacked Boss. Java's HP is 100/100. Boss's HP is 15/40.
There are no Minions to attack.
Boss spawned Minion (HP 10, AT 25, DF 0)
1. Attack
2. Charge
3. Shield
Python, what would you like to do? 1
1. Minion (HP 10)
Python, who would you like to attack? 1
Python attacked Minion. Python's HP is 100/100. Minion's HP is 0/10.
Minion has been defeated.
1. Attack
2. Charge
3. Shield
Java, what would you like to do? 1
1. Boss (HP 15)
Java attacked Boss. Java's HP is 100/100. Boss's HP is 5/40.
There are no Minions to attack.
Boss spawned Minion (HP 10, AT 25, DF 0)
1. Attack
2. Charge
3. Shield
Python, what would you like to do? 1
1. Minion (HP 10)
Python, who would you like to attack? 1
Python attacked Minion. Python's HP is 100/100. Minion's HP is 0/10.
Minion has been defeated.
1. Attack
2. Charge
3. Shield
Java, what would you like to do? 1
1. Boss (HP 5)
Java attacked Boss. Java's HP is 100/100. Boss's HP is 0/40.
Boss has been defeated.
You win!

```

Example 2:

```

How many players? 1
Player 1 Name: Python
Boss spawned Minion (HP 5, AT 25, DF 0)
1. Attack
2. Charge
3. Shield
Python, what would you like to do? 2
Python charged.
Minion attacked Python. Minion's HP is 5/5. Python's HP is 80/100.
Boss spawned Minion (HP 5, AT 25, DF 0)
1. Attack
2. Charge
3. Shield
Python, what would you like to do? 2
Python charged.
Minion attacked Python. Minion's HP is 5/5. Python's HP is 60/100.
Minion attacked Python. Minion's HP is 5/5. Python's HP is 40/100.
Boss spawned Minion (HP 5, AT 25, DF 0)
1. Attack
2. Charge
3. Shield
Python, what would you like to do? 1
1. Minion (HP 5)
2. Minion (HP 5)
3. Minion (HP 5)
Python, who would you like to attack? 3
Python attacked Minion. Python's HP is 40/100. Minion's HP is 0/5.
Minion has been defeated.
Minion attacked Python. Minion's HP is 5/5. Python's HP is 20/100.
Minion attacked Python. Minion's HP is 5/5. Python's HP is 0/100.
Python has been defeated.
You lose.

```

Example 3:

```
How many players? 2
Player 1 Name: Python
Player 2 Name: Java
Boss spawned Minion (HP 10, AT 25, DF 0)
1. Attack
2. Charge
3. Shield
Python, what would you like to do? 3
Python is safe from the next 3 attacks.
1. Attack
2. Charge
3. Shield
Java, what would you like to do? 3
Java is safe from the next 3 attacks.
Minion attacked Java. Minion's HP is 10/10. Java's HP is 100/100.
Boss spawned Minion (HP 10, AT 25, DF 0)
1. Attack
2. Charge
3. Shield
Python, what would you like to do? 1
1. Minion (HP 10)
2. Minion (HP 10)
Python, who would you like to attack? 1
Python attacked Minion. Python's HP is 100/100. Minion's HP is 0/10.
Minion has been defeated.
1. Attack
2. Charge
3. Shield
Java, what would you like to do? 1
1. Minion (HP 10)
Java, who would you like to attack? 1
Java attacked Minion. Java's HP is 100/100. Minion's HP is 0/10.
Minion has been defeated.
Boss spawned Minion (HP 10, AT 25, DF 0)
1. Attack
2. Charge
3. Shield
Python, what would you like to do? 4
Invalid input.
1. Attack
2. Charge
3. Shield
Python, what would you like to do? one
Invalid input.
1. Attack
2. Charge
3. Shield
Python, what would you like to do? 1
1. Minion (HP 10)
Python, who would you like to attack? 2
Invalid input.
1. Minion (HP 10)
Python, who would you like to attack? 1
Python attacked Minion. Python's HP is 100/100. Minion's HP is 0/10.
Minion has been defeated.
1. Attack
2. Charge
3. Shield
Java, what would you like to do? 2
Java charged.
There are no Minions to attack.
Boss spawned Minion (HP 10, AT 25, DF 0)
1. Attack
2. Charge
3. Shield
Python, what would you like to do? 1
1. Minion (HP 10)
Python, who would you like to attack? 1
Python attacked Minion. Python's HP is 100/100. Minion's HP is 0/10.
Minion has been defeated.
1. Attack
2. Charge
3. Shield
Java, what would you like to do? 1
1. Boss (HP 40)
Java attacked Boss. Java's HP is 100/100. Boss's HP is 15/40.
There are no Minions to attack.
Boss spawned Minion (HP 10, AT 25, DF 0)
1. Attack
2. Charge
3. Shield
Python, what would you like to do? 1
1. Minion (HP 10)
Python, who would you like to attack? 1
Python attacked Minion. Python's HP is 100/100. Minion's HP is 0/10.
Minion has been defeated.
1. Attack
2. Charge
```

```

3. Shield
Java, what would you like to do? 1
1. Boss (HP 15)
Java attacked Boss. Java's HP is 100/100. Boss's HP is 5/40.
There are no Minions to attack.
Boss spawned Minion (HP 10, AT 25, DF 0)
1. Attack
2. Charge
3. Shield
Python, what would you like to do? 1
1. Minion (HP 10)
Python, who would you like to attack? 1
Python attacked Minion. Python's HP is 100/100. Minion's HP is 0/10.
Minion has been defeated.
1. Attack
2. Charge
3. Shield
Java, what would you like to do? 1
1. Boss (HP 5)
Java attacked Boss. Java's HP is 100/100. Boss's HP is 0/40.
Boss has been defeated.
You win!

```

IndividualProgramming3

Criteria	Ratings			Pts
Gameplay	11 pts Full Marks Programmer fully addresses all of the elements required of the game. Players are prompted for number of players, player names, and player turn actions. The program uses lists to keep track of "alive" characters and the class methods to keep track of character statuses. Printouts are generated to make it clear exactly what is happening at every step of the game. "You win" or "You lose" are printed out at the end of each game. Printouts match the three examples. Users are re-prompted for an input if an invalid input is entered.	5.5 pts Half Marks Programmer mostly addresses all of the elements required of the game. However, some required elements are missing. There may be some logical errors or printouts do not match the three examples in the prompt.	0 pts No Marks	11 pts
Classes	4 pts Full Marks Programmer creates the required Character, MajorCharacter, Boss, and Minion classes and corresponding methods (with capitalization matching exactly the assignment prompt). Methods print output as required for the game. Programmer appropriately uses class inheritance.	2 pts Half Marks Programmer creates the Character, MajorCharacter, Boss, and Minion classes, but there are some flaws to the class methods and/or initialization. Methods might not be written for the appropriate class as required by the prompt.	0 pts No Marks	4 pts
Loops	2 pts Full Marks Programmers use loops to and lists to handle repeated tasks		0 pts No Marks	2 pts
Readability	3 pts Full Marks Code is well-commented and easy to follow. Variables are appropriately named.		0 pts No Marks	3 pts

Total Points: 20