UMKC School of Computing and Engineering

Assignment 3

Due: Sunday October 8

Battle Game

Review Topics:

- Classes
- Inheritance
- Functions
- Pass by reference
- Random numbers
- Control flow

About:

Write a program where two characters battle, choosing what kind of attack to do each round, and taking damage each round. When one of the characters hits 0 HP, the game is over.

Utility Functions:

Create a set of files, Utils.hpp and Utils.cpp . Within these files you need to declare and define:

GetUserInput(...)

Parameters	**Return Value**
• minimum, int • maximum, int	Integer

Prompt the user for input (as an integer), and validate whether it is between the valid *minimum* and *maximum* values. Only return the user's choice once they enter a valid option.

GetRandom(...)

Parameters	**Return Value**
• minimum, int • maximum, int	Integer

Randomly generate a number between *minimum* and *maximum*.

Objects:

You will write three objects for this game: Character, Enemy, and Player. Both Enemy and Player will inherit from Character, and Character will contain the generic code that both Enemy and Player can use. Player and Enemy will have special code, based on whether they're user-controlled or “AI”-controlled.

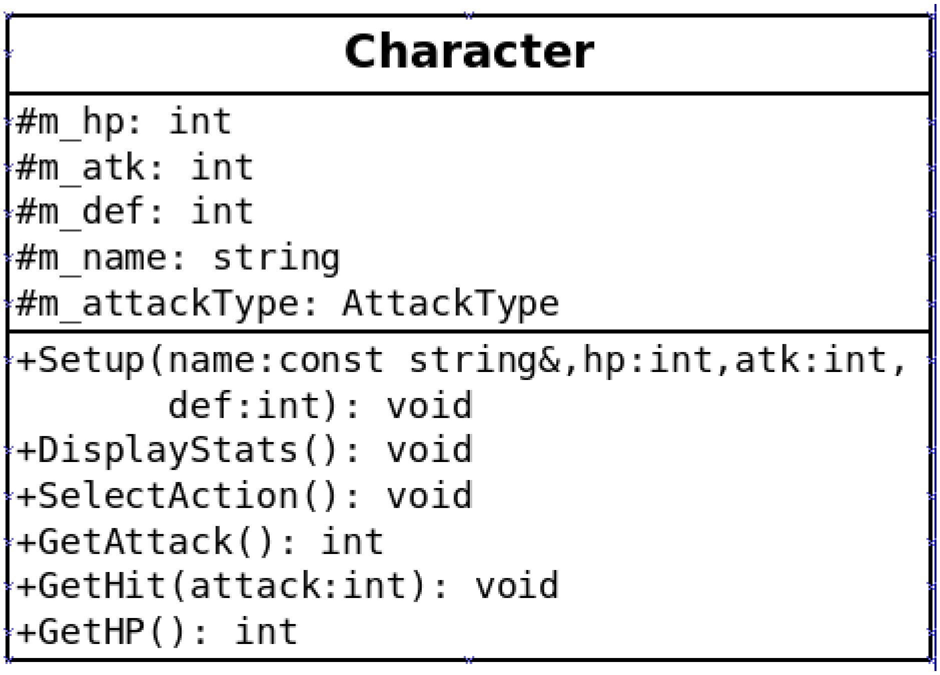
Remember to store each class in their own files - .hpp for declarations, and .cpp for definitions.

Character

First, in the Character.hpp file, declare an **enum** called `AttackType`.

It should have two values: OFFENSIVE, 1, and DEFENSIVE, 2.

This enum will be used within the Character class.



```
Character
#m_hp: int
#m_atk: int
#m_def: int
#m_name: string
#m_attackType: AttackType
+Setup(name:const string&,hp:int,atk:int,
       def:int): void
+DisplayStats(): void
+SelectAction(): void
+GetAttack(): int
+GetHit(attack:int): void
+GetHP(): int
```

# = protected members
+ = public members

Public Members:

Setup(...)

Parameters

- name, const string reference
- hp, int
- atk, int
- def, int

Return Value

None

This function initializes the member variables m_hp, m_atk, m_def, and m_name, with the values passed in.

DisplayStats()

Parameters	Return Value
None	None

Utilize `cout` to display the character's name, HP, ATK, and DEF.

SelectAction()

Parameters	Return Value
None	None

This function will be empty for the Character class.

GetAttack()

Parameters	Return Value
None	Integer

This function will calculate how much damage the character will do, based on their attack type and their ATK stat.

If the character chose an OFFENSIVE attack, add a *random value* (Use the GetRandom function) between 1 and 3 to the attack stat and return that value. Otherwise, just return the attack stat.

You might want to output a message displaying the character's name and the type of attack or damage they're doing.

GetHit(...)

Parameters	Return Value
• attack, int	Void

This function will take the input of attack damage, which was generated by the opposing character's `GetAttack()` function.

If the character is doing an OFFENSIVE attack, damage will be the `attack` parameter value minus the character's defense (m_def).

If the character is doing a DEFENSIVE attack, damage will be the `attack` parameter, minus the defense stat, minus a random value between 1 and 3 (Use the GetRandom function). Make sure that the damage value is greater than or equal to zero before subtracting it from the character's HP.

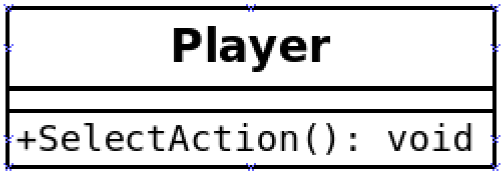
GetHP()

Parameters	Return Value
None	Integer

Return the value of m_hp.

Player

The Player class inherits from the Character class. It only overwrites the SelectAction function.



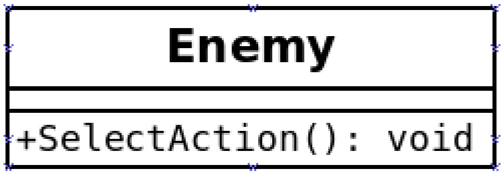
SelectAction()

Parameters	Return Value
None	None

For the Player class, this function will prompt the user to enter an option (1 for offensive, 2 for defensive). Based on their answer, the m_`attackType` will be set, to prepare the player for their `GetAttack()` / `GetHit(...)` functions.

Enemy

The Player class inherits from the Character class. It only overwrites the SelectAction function.



SelectAction()

Parameters	**Return Value**
None	None

For the Enemy class, this function will randomly generate a choice for the enemy (1 for offensive, 2 for defensive). Based on their answer, the m_`attackType` will be set, to prepare the player for their `GetAttack()` / `GetHit(...)` functions.

Additional Functionality:

This function will be within the main.cpp file (or whatever file stores main).

SetupCharacters(...)

Parameters

- player, a Character reference
- enemy, a Character reference

Return Value

None

For this function, it will first ask the user to enter their name. Secondly, it will display a set of three stats to choose from:

Option	Attack	Defense
1	5	15
2	15	5
3	10	10

The user selects the stats for the player, and the enemy's stats are randomly selected. Use the `Setup(...)` function from the Character class to initialize both characters.

Program Flow:

1. In the main program, create a **Player** and **Enemy** object.

2. First Loop: Character Creation

   1 Call the `SetupCharacters(...)` function to set up the player and enemy's beginning stats.

   2 Afterwards, display both the player and enemy's stats (via `DisplayStats()` function), and ask them if this is OK. (Use a number menu, so you can use `GetUserInput(...)`).

   3 Keep looping through the stat creation until the player chooses the “ready” option.

3. Second Loop: Gameplay – Continue looping until one character's life is less than or equal to 0.

   1 Display the round # and both the character's stats at the beginning of each round.

   2 Display the action menu:

      1. Offensive Attack

      2. Defensive Attack

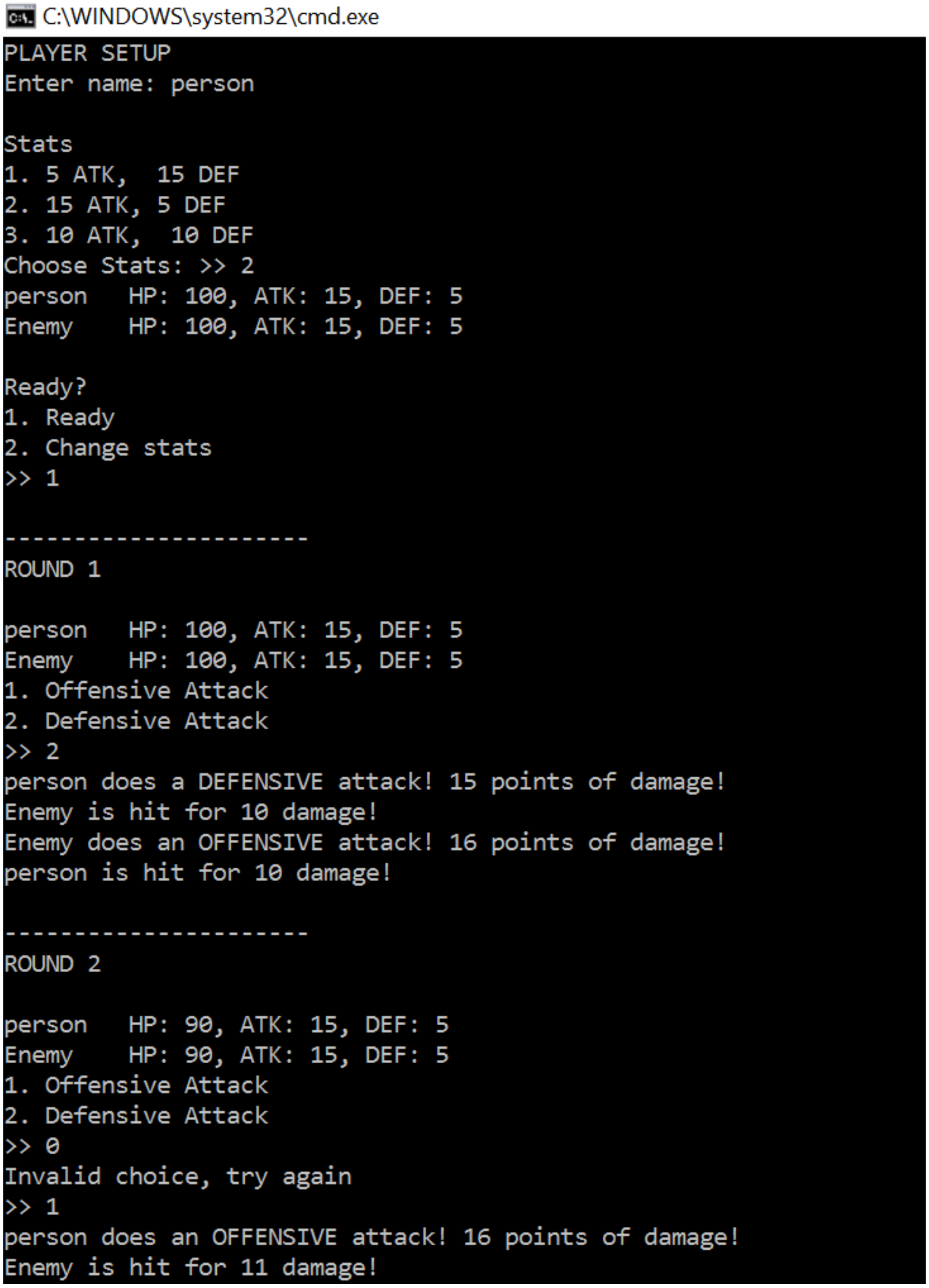
   3. Call each character's `SelectAction()` function.

   4. Afterwards, call both character's `GetHit(...)` functions, passing in the attack value (from `GetAttack()`) from the opposite character.

4. Outside of the loop – game over

   1 Check to see whose HP is <= 0. Display “You Win” if the player is still alive, and “You Lose” if the player has died.

Here is a sample of the output:



C:\WINDOWS\system32\cmd.exe

```
PLAYER SETUP
Enter name: person

Stats
1. 5 ATK,  15 DEF
2. 15 ATK, 5 DEF
3. 10 ATK,  10 DEF
Choose Stats: >> 2
person   HP: 100, ATK: 15, DEF: 5
Enemy    HP: 100, ATK: 15, DEF: 5

Ready?
1. Ready
2. Change stats
>> 1

----------------------
ROUND 1

person   HP: 100, ATK: 15, DEF: 5
Enemy    HP: 100, ATK: 15, DEF: 5
1. Offensive Attack
2. Defensive Attack
>> 2
person does a DEFENSIVE attack! 15 points of damage!
Enemy is hit for 10 damage!
Enemy does an OFFENSIVE attack! 16 points of damage!
person is hit for 10 damage!

----------------------
ROUND 2

person   HP: 90, ATK: 15, DEF: 5
Enemy    HP: 90, ATK: 15, DEF: 5
1. Offensive Attack
2. Defensive Attack
>> 0
Invalid choice, try again
>> 1
person does an OFFENSIVE attack! 16 points of damage!
Enemy is hit for 11 damage!
```

The game will continue till one of the players win/lose.

Program Submission:

Zip up the **.cpp**, **.hpp (or .h)**, and **.txt** files and submit them via Blackboard under Program 3 before the due date.