

# 16.6 Homework 6

## Introduction

Students will create a C++ program that simulates a Pokemon battle as in Homework 5. This time, students will be expected to create a Pokemon trainer that owns 3 pokemons and an enemy pokemon using object-oriented programming (OOP). The assignment is focused on dynamic variables and composition.

## Scenario

You've been assigned the role of creating a Pokemon fight simulator between a Pokemon Trainer's Pokemons (player) and the enemy CPU's Mewtwo.

The player being the pokemon trainer will form its team based on a list of pokemons that will be provided in a file.

After the team is created, you need to create a Pokemon battle simulation that will run until the health points (HP) of the pokemons in the team or the enemy reaches 0.

In the simulation, the player's Pokemons will battle the enemy Mewtwo. Mewtwo will always be the first one to initiate the attack, then you'll attack, and so on until the simulation ends.

The pokemon trainer can choose the attack of the current pokemon fighting. Once a pokemon's HP reaches 0 you cannot use it anymore.

Once the battle is over, you will be greeted with the message "You win" or "You lose" depending on whether the pokemon trainer won the battle.

## Instructions to complete the assignment

Your code must perform these major operations:

- Utilize a while loop to continuously get each person's turn (player and CPU)
- Use OOP to create a Pokemon object, allowing 3 attacks which are to be made using functions
- Use OOP to create a Pokemon Trainer that has an array of pokemons. The pokemons will be provided from the input
- Using print statements that reflect the status of the battle

This Homework is a follow up from Homework 5, that means you already have Move.h, Pokemon.h, and Pokemon.cpp. We are going to made minor changes to Pokemon.h, and Pokemon.cpp.

Pokemon.h and Pokemon.cpp:

Delete:

- Private attribute *isConfused*
- Accessor *getIsConfused()*
- Mutator *setIsConfused(bool)*

Modify:

- Constructor *Pokemon(string name)*: The class constructors takes in name as parameter and creates the pokemon object accordingly with the right values for health and available attacks (details in the list below). Note that Mewtwo only gets one attack, so 2 elements of the Moves array will remain empty.
- function *move(int index, Pokemon& target)*: This function uses as parameter the index of the attack to use (from 0 to 2) and a reference to the pokemon who is being attacked. The health of the target will be reduced according to the attack received. You won't confuse your target in this simulation.

Add:

- *Default constructor*: Will set health equal to zero, name equal to empty string.
- Accessor *getName*: returning the Pokemon name.
- Accessors *getFirstMove*, *getSecondMove*, and *getThirdMove* that will return Pokemon's moves 1, 2, and 3 respectively.

Pokemon Stats

- Pokemon different than Mewtwo (270 HP)
  - 3 Moves provided in the input file
- Mewtwo (650 HP)
  - Psycho Cut (-90 HP)

PokemonTrainer.h and PokemonTrainer.cpp:

Private attributes:

- *maxNumberPokemons* (int)
- *currentNumberPokemons* (int)
- *indexPokemonFighting* (int)
- *teamHealth* (int)
- *pokemons* (pokemon\*)

Public member functions:

- *Default constructor*: will set the maximum number of pokemons to 3, the current number of pokemons to 0, the index of the pokemon fighting to 0, the team health to 0, and will create the array of pokemons with the maximum number of pokemons as size;
- *Copy constructor* (make sure to create a deep copy)
- *Destructor*
- Overloaded assignment *operator =* (make sure to create deep copy).
- Mutator *setTeamHealth*

- Accessors *getTeamHealth*, *getMaxNumberPokemons*, *getCurrentNumberPokemons*, and *getIndexOfCurrentPokemonFighting*
- void type *addPokemon*: Takes a pokemon (Pokemon) as parameter and add it to the team of pokemos. Increase the number of pokemons and if the team is full do not add it.
- void type *displayTeam*: Presents the team with the format

Pokemon Team

```
<Pokemon 1 Name> with moves <Pokemon 1 moves>
<Pokemon 2 Name> with moves <Pokemon 2 moves>
<Pokemon 3 Name> with moves <Pokemon 3 moves>
```

- Pokemon& type *getPokemonFighting*: returns the current pokemon fighting
- *getPokemonFightingName*: returns the name of the current pokemon fighting
- *getPokemonFightingHealth*: returns the health of the current pokemon fighting
- void type *displayPokemonFightingMoves*: displays the moves of the current pokemon fighting
- *getSelectedMoveIndex*: Takes a move name (string) as a parameter, looks for the move in the pokemon moves and return the corresponding index.
- void type *pokemonFightingAttacks*: takes in as parameter the index of the attack to use (from 0 to 2) and a reference to the pokemon who is being attacked. The current pokemon fighting attacks with the corresponding move.
- void type *nextPokemonToFight*: increases indexPokemonFighting.

main.cpp

- Read the name of the file with the list of pokemons from the input.
- Create an array of 5 pokemons. There will always be 5 pokemons in the file.
- Read pokemons from file and store them in the array
- Create Pokemon Trainer object and enemy Pokemon Mewtwo
- Display pokemons list so the user can form their team
- Form the team
- Start the battle. End if either Mewtwo or the team of pokemons HP reaches 0
  - mewtwo always attacks first
  - check if we need to change pokemons, if so print the change in pokemon
  - print current pokemon and attack options
  - get user move choice
  - attack based on the user choice
- Print who won the battle

Input file format example.txt:

```
Pikachu // Pokemon 1 info
Thunderbolt
125
Electro Ball
50
Quick Attack
```

90

...

```
Bulbasaur      // Pokemon 5 info
Tackle
60
Vine Whip
85
Sludge Bomb
150
```

**\*\*Console Input/Output \*\***

**Sample input**

```
PokemonsList.txt
1
2
3
Thunderbolt
Quick Attack
Ember
Flame Burst
Flame Burst
Vine Whip
Sludge Bomb
```

**Sample output**

```
List of Pokemons to choose
Number 1
Name: Pikachu
Moves: Thunderbolt, Electro Ball, or Quick Attack
Number 2
Name: Charmander
Moves: Scratch, Ember, or Flame Burst
Number 3
Name: Bulbasaur
Moves: Tackle, Vine Whip, or Sludge Bomb
Number 4
Name: Squirtle
Moves: Tackle, Bubble, or Water Pulse
Number 5
Name: Spearow
Moves: Peck, Quick Attack, or Body Slam
```

Enter the number of the pokemon you want to add to your team:

Enter the number of the pokemon you want to add to your team:  
Enter the number of the pokemon you want to add to your team:

Pokemon Team

Pikachu with moves Thunderbolt, Electro Ball, or Quick Attack  
Charmander with moves Scratch, Ember, or Flame Burst  
Bulbasaur with moves Tackle, Vine Whip, or Sludge Bomb

Prepare to fight, the battle will start now!

Mewtwo used Psycho Cut

Your current pokemon fighting is Pikachu

It can use Thunderbolt, Electro Ball, or Quick Attack

Pikachu used Thunderbolt

Mewtwo used Psycho Cut

Your current pokemon fighting is Pikachu

It can use Thunderbolt, Electro Ball, or Quick Attack

Pikachu used Quick Attack

Mewtwo used Psycho Cut

Pikachu can't keep fighting

Charmander prepares

Your current pokemon fighting is Charmander

It can use Scratch, Ember, or Flame Burst

Charmander used Ember

Mewtwo used Psycho Cut

Your current pokemon fighting is Charmander

It can use Scratch, Ember, or Flame Burst

Charmander used Flame Burst

Mewtwo used Psycho Cut

Your current pokemon fighting is Charmander

It can use Scratch, Ember, or Flame Burst

Charmander used Flame Burst

Mewtwo used Psycho Cut

Charmander can't keep fighting

Bulbasaur prepares

Your current pokemon fighting is Bulbasaur

It can use Tackle, Vine Whip, or Sludge Bomb

Bulbasaur used Vine Whip

Mewtwo used Psycho Cut

Your current pokemon fighting is Bulbasaur

It can use Tackle, Vine Whip, or Sludge Bomb

Bulbasaur used Sludge Bomb

Mewtwo's health points reached 0. You win!