# Pokemon Programming Project

Today you will begin constructing a model of the Pokemon game you may have played at home or with friends. Object-oriented programming uses data to model real-world objects and situations. In this exercise, you will use objects and classes to model the game sequence experienced in Pokemon (which is itself a representation of quests, battles, growth, and evolution)!

You may use your textbook, notes, Tricky Code Cheat Sheet, 4 Commandments of Scope, and any classroom posters or examples to help you with this project. The only online resources you may use for this assignment are Bulbepedia or Nintendo’s official Pokemon website. You should work independently on writing this code.

Toward the end of class, your instructor will ask you to trade papers with another student. Write your name on their paper (and be sure that they write their name on yours!) so that you both get credit for helping error-check each others’ work.

## Exercise 1

(A) In the space below, create a Pokemon class that meets the following requirements:

* Uses a field to declare a variable of the type boolean to indicate whether the pokemon isFishing or isFighting
* Uses a field to declare a variable of type int to store hitPoints
* Contains a constructor for new instance of the class by passing initial isFishing, isFighting, and hitPoints values as parameters
* Contains an instance method damage that subtracts amountOfDamage from hitPoints (Hint: this should be a mutator method.)

NAME: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ERROR-CHECKER: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(B) Explain the difference between a field and a parameter. What is the difference in their syntax? What is the difference in their scope and in the ways in which they may be used?

## Exercise 2

Write client code that creates 2 instances of *Pokemon* called charmander and raichu. (Hint: you may want to visit the Pokemon stats page for your initial values. <http://tinyurl.com/no4mzlc> ) Include comments to explain your code and your design choices.

## Exercise 3

(A) Write client code that compares each of the *Pokemon’s* hit points using an accessor. Then, write a method that outputs the results to the console. TIP: Remember that you must write your own toString method since this is a customized class.

(B) What is the difference between an accessor and a mutator? What naming conventions are used with accessor and mutators?

(C) How do you write a class whose objects can easily be printed on the console?

## Exercise 4

(A) According to the Bulbepedia wiki, every stat for every instance of *Pokemon* should always be less than 255. At no point will the total of a pokemon’s stats ever exceed 510 points. How could you ensure that client methods never incorrectly alter a Pokemon’s stats? Explain and give an example below.

(B) Are class invariants inherent to code? Or can they be changed? How does a coder decide whether to enforce a class invariant? How does a coder decide what the invariant should be?

## Exercise 5

Create a class called Trainer that represents a player’s name. The class should have fields representing the player’s first name, last name, middle initial, and whether they have ever played the game before. Your class should only contain fields for now.