

Background: Mr. Burns doesn't care about almost anything ... but himself and his money. So, when his power plant leaked radioactive goo that caused several species of wildlife to go extinct, he was only concerned with the public perception as it might effect his income and possible jail time. After the appearance of the three-eyed fish, Blinky, other rumors surfaced around the Springfield Nuclear Power Plant. One of them is high concern over the mutation rate of the rare Springfield molted platypus. With barely more than 50 left in the wild, the word "extinction" has been tossed around. So, to quell the media, Mr. Burns had 30 of them captured, dissected, and analyzed to check for signs of mutation. He found that the mutation rate is 2% each month, but when they do mutate, they become sterile and cannot reproduce. With this information, he wants to create one of those newfangled computer simulations that the press loves so much. That's where you come in!

Specifications: In this assignment, you will create a class called Platypus. Below, we will describe what will define a Platypus. You will also create a main function in which you will create objects of type Platypus to test the functionality of your new user-defined type. In the driver, you will create enough objects of your new type(s) in order to adequately test their functions. We leave it up to you to do this properly.

Your Platypus class is to contain the following:

Member variables:

- a float for weight
- a short for age (months)
- a string for name
- a char for gender
- a bool to indicate whether *alive* (or not)
- a bool to indicate whether *mutant* (or not)

Member functions:

- a default constructor that creates a dead Platypus
- a constructor that you can pass values to so as to establish its gender, weight, age, and name; it will default to *alive* and *not mutant*.
- a *print* function that will output to the screen the attributes of that Platypus in a nice, easy to read format.
- an *ageMe* function that returns nothing but increments the object's age. It will also include a 2% chance that the object will become a mutant. Further, the Platypus has a chance of becoming dead each time it ages. This chance is ten times the Platypus' weight. A 5 pound Platypus has a 50% chance of death. A 10 pound Platypus (or heavier) has a 100% chance of death.

- a fight function that accepts another Platypus object as a parameter. It will have the calling Platypus attack the other (passed in) Platypus. The survivor is based on a "fight ratio": it is calculated as

$(\text{calling_platypus_weight} / \text{other_platypus_weight}) * 50.$

A random value from 1 to 100 is obtained. If it is less than the fight ratio, the calling Platypus survives; otherwise the other platypus survives.

- an eat function that increases the weight of the Platypus by a random amount from 0.1% to 5.0% of the Platypus' current weight.
- A *hatch* function that will randomly set up a newborn Platypus with alive=true, mutant=false, and age=0. Gender will randomly be 'm' or 'f' with equal probability. Weight will randomly be between 0.1 and 1.0 pounds. Name will be randomly chosen from one of two files. The [m_names.txt](#) file contains names for male Platypuses (Platypi?). The [f_names.txt](#) file contains names for female Platypuses. (Hint: Write a function that accepts either 'm' or 'f' and returns a random name from one of those files.)

Note: Think very carefully about writing the above functions and how they should be used. There are, indeed, circumstances when some functions should not execute. For example, a dead platypus shouldn't eat anything.