Java Fundamentals
7-2: Parameters and Overloading Methods
Practice Activities

Lesson Objectives:

- Use access modifiers
- Pass objects to methods
- Return objects from methods
- Use variable argument methods
- Overload constructors
- Overload methods
- Write a class with specified arrays, constructors, and methods

Vocabulary:

Identify the vocabulary word for each definition below.

public	A type of access modifier. Permits access from anywhere.
Constructor	Used to assign initial values to instance variables of a class.
varargs	A way to call a method with a variable number of arguments.
Constructor Overloading	Having more than one constructor or method with the same name but different arguments.
private	A type of access modifier. Permits access only from inside the same class.
default constructor	A constructor that does not have any parameters.
Access Modifier	Used to specify accessibility for variables, methods, and classes.

Try It/Solve It:

1. Create a segment of code that initializes a public class *Fish.* Let the class contain a String *typeOfFish,* and an integer *friendliness.* Do not set values to these variables yet. These are instance variables and will be set inside the class constructors.

2. Create a public constructor (a method with the same name as the class) inside the class *Fish*. This constructor should take in no arguments. Inside the constructor, set *typeOfFish* to "Unknown" and *friendliness* to 3, which we are assuming is the generic friendliness of fish.

3. Create another public constructor inside the class *Fish.* Have this constructor take in a string *t* and an integer *f. Let typeOfFish* equal *t*, and *friendliness* equal *f.*

4. Explain why is is possible to have more than one constructor with the same name and different arguments.

   It is possible due to method overloading which allows multiple methods with the same name to exist as long as their parameter lists are different.

5. Create a method inside the class *Fish* called *getFriendliness(),* which takes in no arguments and returns the friendliness level of the fish.

6. Write a segment of code that initializes 2 new fish as defined below:
   a. Fish 1: Name – *Amber*, Type – *AngelFish,* Friendliness level – 5 (very friendly)
   b. Fish 2: Name – *James*, Type – *Guppy*, Friendliness level – 3 (neutral)

7. Create a method *nicestFish* that takes in two fish as parameters, compares the friendliness level of two fish, and returns the fish with the higher friendliness. Test this method with the fish defined in problem 6. (Friendliness scale: 1 mean, 2 not friendly, 3 neutral, 4 friendly, 5 very friendly) *Hint: fishName*.getFriendliness() gives you the integer number of the friendliness of *fishName.* You have already created *getFriendliness()* in problem 5.

8. Modify the method *nicestFish()* to take be a variable argument method that takes in a variable number of fish and returns the nicest fish out of the fish it is given. *Hint:* Inside of the method, create a new fish called *temp*. Set *temp* equal to the first fish passed into the method. Use a for loop to go through all the fish passed into the method and if you discover a fish that is more friendly than *temp*, set *temp* equal to that fish. After the for loop is complete, *temp* should be the friendliest fish. Return *temp*.

9. Test your method *nicestFish()* with the fish described in problem 6. Which fish is returned?

10. Determine the best access modifier for each of the following situations:
    a. A class Employee records the name, address, salary, and phone number.
    b. An adding method inside of a class BasicMath that is also used in the Algebra class.

    A.The best modifier for the fields(name, address, salary and phone number) in the Employee class is 'private'.

    B. The best modifier to used for the adding method 'BasicMath' is 'public'.