**Java**

**Dogs!**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**PART 1**

**1)** Create a class called Dog

With these instance variables: int age, String name

This constructor: Dog() - sets age to 0, and name to "no name yet"

And these methods:

int increaseAge() - adds one to the age, returns the new age

boolean setName(String newName) - if the current name is "no name yet, change the name to newName and return true. If the current name is not "no name yet"

then don't change the name and return false

String name() - returns the name of the Dog

int age() - returns the age of the Dog

**2)** Create a class called LookAtAllMyDogs that has a main method.

**3)** Create an array of 5 Dogs.

**4)** Use a for loop to put a new Dog in each spot in the array by calling the constructor of Dog.

**5)** Give the user these options - Print out the array of Dogs before giving the user a choice. Print each Dog's index - name - age

**1)** Add one year - call the increaseAge() method on all of the Dogs

**2)** Name a Dog - ask for the index, then the new name. Print out "Name changed" or

"That Dog has already been named"

**PART 2**

In your AllMyDogs program:

**1)** Add the option to remove a Dog. Ask the user for the index, set that spot in the

array to null

**2)** Since you can remove Dogs now, you need to update your way of printing out the

list of Dogs. If a particular spot in the array points to null, print out "empty"

**Then write me some comments explaining these parts of your program:**

**a)** Naming a Dog

**b)** Increasing a Dog's age

**c)** Giving a Dog away

**d)** Printing without crashing

Each line should have two parts to its comment:

1) what the code does 2) why you need that line of code

For example:

/\* 1) Creates a Scanner connected to the console saved by the variable sc

\* 2) Allows the user to input information

\*/

Scanner sc = new Scanner(System.in)

**Part 3**

**In the Dog class:**

**1)** Add another constructor to the Dog class that accepts a String for the name, and an int for the age as parameters.

**2)** Add a method (remember they are not real dogs) that makes your Dog die of old age.

This method should be called checkOldAge() and return a boolean that is false if

the Dog should keep living.

Using a random number and the Dog's current age to decide if the method should

return true or false. The probability of the Dog dying should increase as its age

increases.

**In the LookAtAllMyDogs class:**

**3)** Add a menu option to adopt an older Dog. This is where you use your new constructor.

The user should not be able to adopt a Dog if the array has no empty spaces.

The user should not enter an index for the new Dog, the program should automatically

place the Dog in the first available space.

**4)** Every time all of the Dogs get older, call checkOldAge() for each of the Dogs. If it returns true for any of them, remove that Dog from the array and print out something nice to remember the Dog by.

**Part 4**

**1)** What if you want more Dogs? If the user wants to add a Dog and the array is already full, make the array bigger and add the Dog.

This is going to take a little thinking because you cannot really change the size of an array, but you can make a new one.

**2)** Dogs need buddies.

Add a method to the Dog class to set a Dog's friend that takes a Dog as a parameter, you may also need to add a friend instance variable.

Let's assume for this program that if dogA is friends with dogB, then dogB is also friends with dogA. So calling dogA.setFriend(dogB) will make dogA friends with dogB, and dogB friends with dogA.

Then add a method called getFriend() that returns the Dog's friend.

When you print out the list of Dogs, add a :) after the Dog's name if they are next to their friend.