**CS 1302 – Test 3 Information**

**Information about the test**

General

* Test 3 covers Ch 4 & Ch 5.
* You can begin the test approximately 10 minutes before the official beginning of class. When you enter the room, leave everything (all books, phones, smart watches, *etc*)at the front of the room.
* The test is completely writing classes, methods, and snippets.

**Ch 4 Expectations**

1. Know these *ArrayList* methods: add(obj), add(indx,obj), addAll(anotherList), clear, contains(obj), get(indx), indexOf(obj), isEmpty, remove(indx), remove(obj), set(indx,obj), size, constructor: ArrayList<>(anotherList)
2. Write code to override equals.
3. Utilize *ArrayList* methods that rely on *equals*: *contains(obj), indexOf(obj), remove(obj)*.

**Ch 5 Expectations**

1. Write abstract classes and subclasses.
2. Write an interface and implement it.
3. Write code to implement the *Comparable* interface.
4. How to sort an ArrayList of primitives or custom objects.
5. Implement one-to-many using an *ArrayList*, similar to what is in the notes, exercises, and homework:
6. Implement: *add, get, remove*
7. Write methods that utilize a dummy (key) object to search for an object in an *ArrayList* using *contains(obj), indexOf(obj), remove(obj)*.
8. Write methods that process all the items in the array.
9. Write methods that traverse all the items in the array, that use *instance* and casting to filter out subclasses to process or return.

**Note about Format of Test**

* You will be provided a handout that shows a class diagram of a 1-many relationship. For example, a *Person* has many *Animals* (abstract) with subclasses *Dog* and *Cat*.
* The class diagram uses yellow highlight to circle exactly what methods/classes are to be written. You can assume that the methods that are not highlighted are available to you and do not have to be written. These are the questions:
* Almost everything (perhaps everything) on the test is summarized in: (a) Ch. 5, Section 8, (b) Exercises 1, 2, 7, 10 in Ch 5 (c) HW 5. However, the test will have only a single 1-many relationship. For example, from the example in Ch 5, Section 8, just remove the *flyers* list.

**Test Questions**

This is an idea of what the 7 questions on the test look like, using the example from above:

1. (7%) Write the *Animal* class and a method or two
2. (24%) Write the *Dog* class and several methods/constructors
3. (40%) Write the *Person* class and 4 methods (*addAnimal*, *getAnimal, removeAnimal,* and one more that iterates over the animals filters out only the *Dogs* and does something with them).
4. (16%) Use the classes you wrote.
5. (8%) Modify a class so that an arraylist of objects from that class can be sorted according to some requirements. For example, if I give you the code for an *Employee* class and you need to modify it so that an arraylist of *Employee* objects can be sorted on their *employNum*, then there are 3 things you need to do: (a) add: *implements Comparable<Employee>* to the signature of the class, (b), implement the *compareTo* method, (c) override the *equals* method.
6. (5%) Write a *get* method that accepts an argument and the method will use the dummy object approach to find the dummy in a list and do something with it (return it, perhaps). For example, see: (a) Ch 5, Section 3, *getPet(name),* (b) Ch 4, Section 6, *getAccount(accountNumber).*