CS 2510 Exam 2 – Spring 2010

Name: _____

Student Id (last 4 digits): _____

• Write down the answers in the space provided.
• You may use all parts of the Java language we have learned. If you need a method and you don't know whether it is provided, define it. You do not need to include the curly braces for every `if` or every `else`, as long as the statements you write are correct in standard Java.
• For tests you only need to provide the expression that computes the actual value, connecting it with an arrow to the expected value. For example `s.method() -> true` is sufficient.
• This exam focuses on problem solving. The **design recipe** is your key to success. Feel free to add plain English comments explaining what you are trying to do, or what would be the next step as you solve the problem.
• You are *not* required to provide a method template unless the problem specifically asks for one. However, be prepared to struggle if you choose to skip the template step.
• We will not answer *any* questions during the exam.

Good luck.

Problem	Points	/
1		/27
Total		/27

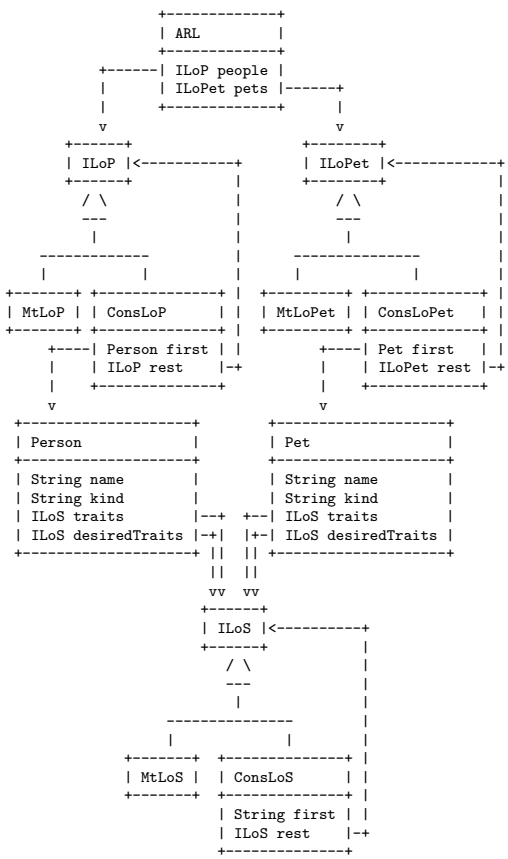
27 Points

Problem 1

We are writing a program for an Animal Rescue League. They find stray and abandoned cats and dogs, take care of them till they are well, then give them to people who would like to adopt them.

To make sure each person is matched with a compatible pet, they would like us to help in finding the matching pets for a person that wants to adopt a pet.

The class diagram shown here describes how the Animal Rescue League (ARL) keeps track of the people and pets. A person wanting to adopt a pet has to wait until a suitable animal becomes available.



A. **Problem analysis and data definition part. [7 points]**

- Make examples of people, pets, lists of people, lists of pets, and the `ARL` class. The `traits` field describes the traits of this person or pet, the `desiredTraits` field describes the traits desired of the matched partner.

  Here is an example of the Animal Rescue League records for today. Turn them into examples.

  Traits recorded (some are pertinent only for animals):

  energetic, calm, outdoorsy, indoors only, kid friendly, no kids, long haired, trained,

  Persons waiting for animals:

  - Tom - is energetic, outdoorsy, has no kids, wants a cat that is energetic, outdoorsy.
  - Ann - is calm, has kids, wants a dog, indoors only, kid friendly, trained.
  - Jen - is calm, wants an outdoor cat.

  Pets waiting to be adopted:

  - Kitty - an energetic, outdoorsy, long haired cat, needing an owner who is energetic, outdoorsy.
  - Doggie - an indoors only, kid friendly, trained dog, wants an owner who is calm.
  - Boots - a kid friendly indoor only cat needing an owner who is calm

- Make one more example of a person and one more example of a pet that are compatible with each other, i.e. the person has the pet's desired traits and the pet has the traits the person desires of the pet.

Note:

You may use the following notation for defining `ConsLo?` lists:

```
ILo?  mylist = new ConsLo?(item1, item2, ..., item-n, mtlo?)
```

Solution [POINTS 7: 2 points for examples of persons and their lists, 2 points for examples of pets and their lists,2 points for the class ARL, 1 point for the new pair of person/pet]

```
String e = "energetic";
String c = "calm";
String o = "outdoorsy";
String i = "indoors only";
String k = "kid friendly";
String n = "no kids";
String l = "long haired";
String t = "trained";

ILoS mtlos = new MtLoS();
ILoS eno =
  new ConsLoS(this.e,
      new ConsLoS(this.n,
          new ConsLoS(this.o, this.mtlos)));
ILoS enot = new ConsLoS(this.t, this.eno);
ILoS enol = new ConsLoS(this.l, this.eno);
ILoS eo =
  new ConsLoS(this.e,
          new ConsLoS(this.o, this.mtlos));
ILoS ck =
  new ConsLoS(this.c,
          new ConsLoS(this.k, this.mtlos));
ILoS ikt =
  new ConsLoS(this.i,
      new ConsLoS(this.k,
          new ConsLoS(this.t, this.mtlos)));
ILoS cx = new ConsLoS(this.c, this.mtlos);
ILoS ox = new ConsLoS(this.o, this.mtlos);
ILoS ix = new ConsLoS(this.i, this.mtlos);
ILoS elo =
  new ConsLoS(this.e,
      new ConsLoS(this.l,
          new ConsLoS(this.o, this.mtlos)));

Person tom =
  new Person("Tom", "cat", this.eno, this.eo);
Person ann2 =
  new Person("Ann", "dog", this.ck, this.ikt);
Person jen =
  new Person("Jen", "cat", this.cx, this.ox);
```

```
Person bob =
  new Person("Bob", "cat", this.eno, this.enot);
Person ann =
  new Person("Ann", "dog", this.enol, this.eno);

ILoP mtlop = new MtLoP();
ILoP bobList = new ConsLoP(this.bob, this.mtlop);
ILoP annList = new ConsLoP(this.ann, this.mtlop);
ILoP personList = new ConsLoP(this.ann, this.bobList);

ILoP allpeople =
  new ConsLoP(this.tom,
    new ConsLoP(this.ann2,
        new ConsLoP(this.jen,
            new ConsLoP(this.bob,
                new ConsLoP(this.ann, this.mtlop)))));

Pet kitty = new Pet("Kitty", "cat", this.eno, this.eno);
Pet doggy = new Pet("Doggy", "dog", this.eno, this.eno);

Pet kitty2 = new Pet("Kitty", "cat", this.elo, this.eo);
Pet doggie = new Pet("Doggie", "dog", this.ikt, this.cx);
Pet boots = new Pet("Boots", "cat", this.ix, this.cx);

ILoPet mtlopet = new MtLoPet();
ILoPet kittyList = new ConsLoPet(this.kitty, this.mtlopet);
ILoPet doggyList = new ConsLoPet(this.doggy, this.mtlopet);
ILoPet petList = new ConsLoPet(this.kitty, this.doggyList);

ILoPet allpets =
  new ConsLoPet(this.kitty,
    new ConsLoPet(this.doggy,
        new ConsLoPet(this.kitty2,
            new ConsLoPet(this.doggie,
                new ConsLoPet(this.boots, this.mtlopet)))));

ARL arl1 =
  new ARL(this.personList, this.petList);
ARL arlService =
  new ARL(this.allpeople, this.allpets);
```

B. **Method definition part. [20 points]**

Design the method `findMatch` for the class `ARL` that consumes person's name and produces a list of compatible pets available for adoption at this time.

Note: You may need several helper methods. Work through the problem one step at a time, completing a step carefully. You may add a simple comment in English, explaining why you need a helper method, or what is it you are trying to do next. Make sure for each method you design it is clear in which class is the method defined.

Make sure we can see how you have worked towards the solution. A well designed couple of first steps count for more than disorganized guess at 'almost correct' answer.

The Design Recipe is your friend here.

Solution [POINTS 20: *solution not provided.*]

. . .