

## 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.

Problem	Points	/
1		/27
<b>Total</b>		/27

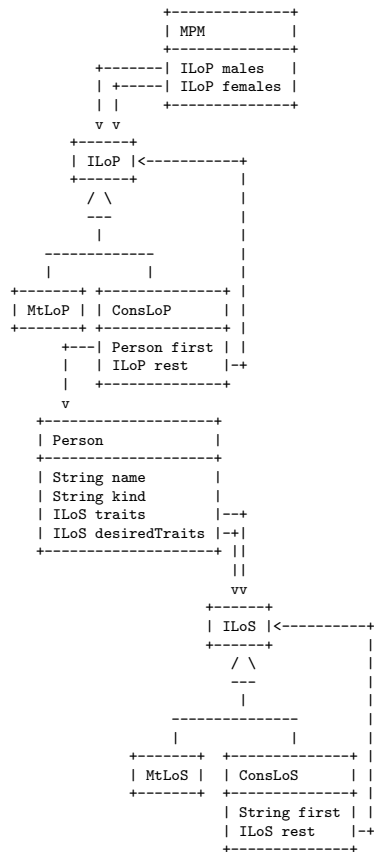
*Good luck.*

**Problem 1**

We are writing a program for the Meet Perfect Match match-making service. They match heterosexual couples, with a guarantee of a perfect match for every customer.

To make sure each person is matched with a compatible person, they would like us to design a program that generates the possible partners for a given customer.

The class diagram shown here describes how the Meet Perfect Match (MPM) keeps track of the people. A person wanting to find a match has to wait until a suitable person becomes available.



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

- Make examples of males, females, lists of males, lists of females, and the MPM class. The `traits` field describes the traits of this person, the `desiredTraits` field describes the traits desired of the matched partner.

Here is an example of the Meet Perfect Partner records for today. Turn them into examples.

*Traits* recorded:

extrovert, introvert, vegan, carnivore, nature lover, sports lover, avid reader, party person

*Males* waiting for partners:

- Bob - is extrovert, nature lover, vegan, wants a partner who is extrovert, nature lover, vegan and a party person.
- Dan - is introvert and avid reader and is looking for a partner who is introvert and a carnivore.
- Ted - is extrovert and sports lover, looking for an extrovert who is an avid reader.

*Females* waiting for partners:

- Kay - is extrovert, nature lover, sports lover, vegan and a party person, looking for an extrovert, nature lover, vegan partner.
  - Bea - is introvert, nature lover, carnivore, looking for an introvert partner who is an avid reader.
  - Meg - is extrovert and sports lover, looking for an extrovert who is an avid reader.
- Make one more example of a male and one more example of a female that are compatible with each other, i.e. the male has the female's desired traits and the female has the traits the male desires of the female.

*Note:*

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

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

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

Design the method `findMatch` for the class `MPM` that consumes person's name and produces a list of compatible persons available for 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.**