

# MPP Midterm Exam 4/11/16

---

Name: \_\_\_\_\_ StudentId: \_\_\_\_\_

There are a total of 35 points possible. Note that there are questions on the back of this page. You have until 12:00 Noon to complete all the questions.

-----

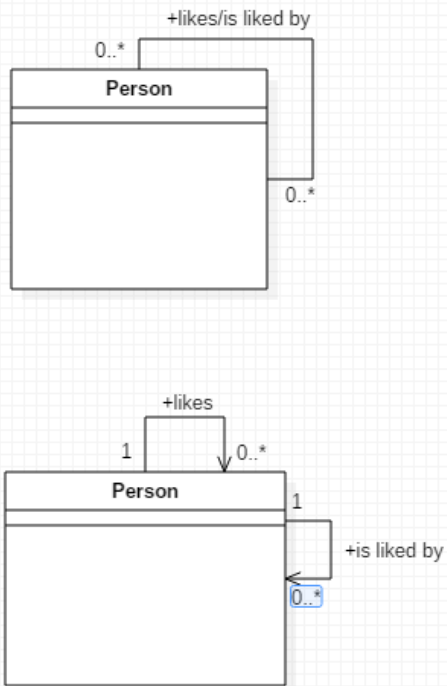
- 1) (4pts) Name two ways of guaranteeing that no user of your class `MyClass` can create a subclass of `MyClass`.

**Solution:**

- a. **Declare `MyClass` to be final**
- b. **Declare the `MyClass` constructor to be private and provide a factory method to provide instances of the class.**

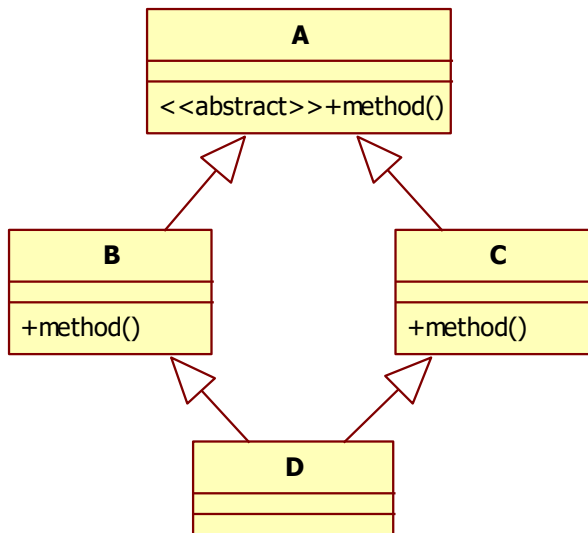
- 2) (4pts) A Person may *like* other Persons, and may also be *liked by* other Persons. Draw a class diagram showing these relationships. Use one or more associations; each should have a name and appropriate multiplicities. You must indicate clearly whether associations are 1-way or 2-way.

**Solution: Both of the following are correct:**



3) (2pts) Describe one advantage of using a static method.  
*You do not have to instantiate the class to use the method.*

4) (2pts) What is the “diamond problem?” Draw a diagram showing the problem.



5) (2pts) What is Java’s approach to handle the “diamond problem?”

*Implementation (a class) can only be 'inherited' / extended once  
Types (interfaces) can be 'inherited' / implemented multiple times*

- 6) (4pts) Name two differences between interfaces and abstract classes in Java (restricted to Java SE 7 and earlier).

*Any 2 of the following are correct:*

*interfaces cannot have implemented methods – abstract classes can*

*interfaces cannot have static methods – abstract classes can*

*interfaces cannot have instance variables – abstract classes can*

*a class can implement multiple interfaces but extend only one abstract class*

- 7) (2pts) What is the Evolving API Problem?

*The Evolving API Problem is the problem that it is not safe to add new methods to an interface in a system that has already been released, because the presence of new methods will break existing implementations.*

- 8) A rectangle can be specified by specifying two sides, but it can also be specified by specifying one side and a diagonal.

A. (2pts) The following code attempts to implement a Rectangle class and provide support for the two ways of constructing a Rectangle. Explain the problem with this code.

```

public class Rectangle {
    double side1, side2, diagonal;
    public Rectangle(double s1, double s2) {
        this.side1 = side1;
        this.side2 = side2;
        diagonal = Math.sqrt(side1 * side1 + side2 * side2);
    }
    public Rectangle(double s1, double diagonal) {
        this.side1 = side1;
        this.diagonal = diagonal;
        side2 = Math.sqrt(diagonal * diagonal - side1 * side1);
    }
    public double computeArea() {
        return side1 * side2;
    }
}

```

*Your Explanation:*

*Solution: The Java compiler does not allow two constructors in a class that have the same signature.*

B. (4pts) Write a pseudo-code solution for the rectangle class for the solving this problem.

*Your rectangle class will now have a private constructor.*

*You will have 2 public static methods that return a Rectangle.*

*For example, -- public static Rectangle createRectangleBySides(double s1, double s2)*

*And -- public static Rectangle createRectangleBySideAndDiag(double s1, double diagonal)*

*These static methods will new a Rectangle and set the sides or side and diagonal in the new Rectangle object. A Rectangle factory can call the static methods based on input provided to it.*

9) (9pts) For the following use case description show the sequence diagram for a developer estimates the remaining hours of development needed for a feature. Assume the developer for this use case is already logged in. Do not show the login use case.

Business rules are:

- i. Remaining development time cannot be a negative number.
- ii. A developer may only estimate his/her features
- iii. For each time a developer estimates the development time we want a WorkLogRecord which will contain the date/time the estimate was entered and the estimated time remaining. WorkLogRecords will be saved in our DataAccess subsystem.

Your sequence diagram must show how you check the above business rules.

*Your answer must show how you check that the development time is non-negative.*

*It also must show how you make sure developers are updating only the features assigned to them. An example approach is:*

- *The developer starts at the developer feature page*
- *The developer requests a list of his/her features*
- *The developer selects one of his/her features*
- *The developer enters in the number of development hours remaining. Show the page validating the entry is non-negative.*

- *The developer clicks submit and a new WorkLogRecord is created with the current date/time and the updated effort time.*
- *The new WorkLogRecord is added to the Feature's list of WorkLogRecords*
- *Save the WorkLogRecord and the Feature to the DataAccess subsystem.*
- *A success page confirms the update*

*See diagram solution midtermSequenceDiagramSoln*