

## CS 2510 Exam 3 – Summer 2012

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

Student Id (last 4 digits): \_\_\_\_\_

- Write down the answers in the space provided.
- You may use all syntax of Java that we have studied in class.
- 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.
- Remember that the phrase “design a class” or “design a method” means more than just providing a definition. It means to design them according to the **design recipe**. 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.

Score		45
-------	--	----

*Good luck!*

**Problem 1**

We've seen multiple instances of objects representing functions, such as predicates and comparison functions. The general idea for representing a function with contract  $A \rightarrow B$  is to use an object with an **apply** method that consumes an  $A$  and produces a  $B$ :

```
// Represents a function from A to B.
interface Fun<A,B> {
    // Apply this function to given argument.
    B apply(A a);
}
```

Design an implementation of `Fun<Posn,Double>` that calculates the distance from the origin to the `posn`. You may assume the following data definition for `Posn` (with the usual constructor):

```
// Represents a point on a Cartesian plane
class Posn {
    Integer x;
    Integer y;
}
```

Hint: the distance between two points is given by  $\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$  and `Math.sqrt` and `Math.sqr` calculate what their names suggest.

**Problem 2**

Assume the following definition for `ListVisitor<X,R>`:

```
// Represents a computation over a list of X producing an R.
interface ListVisitor<X,R> {
    R visitEmpty();
    R visitCons(X first, List<X> rest);
}
```

Design an implementation of `ListVisitor<X,List<Y>>` that maps a given function (of type `Fun<X,Y>`) over a list of `X` to produce a list of `Y`.

**Problem 3**

Design an implementation of `ListVisitor<X,X>` that produces the first element of a non-empty list that minimizes the result of a given `Fun<X,Double>` function. It should signal an error if the list is empty.

For example, if given the distance function from problem 1 and a list of `Posns` representing (4,5), (2,3), and (1,2), it should produce the `posn` representing (1,2) since it is closest to the origin (it minimizes the distance function).

Hint: you should use an accumulator-based design and therefore may need to develop two visitors.

