CS 2510 Exam 1 – Fall 2010
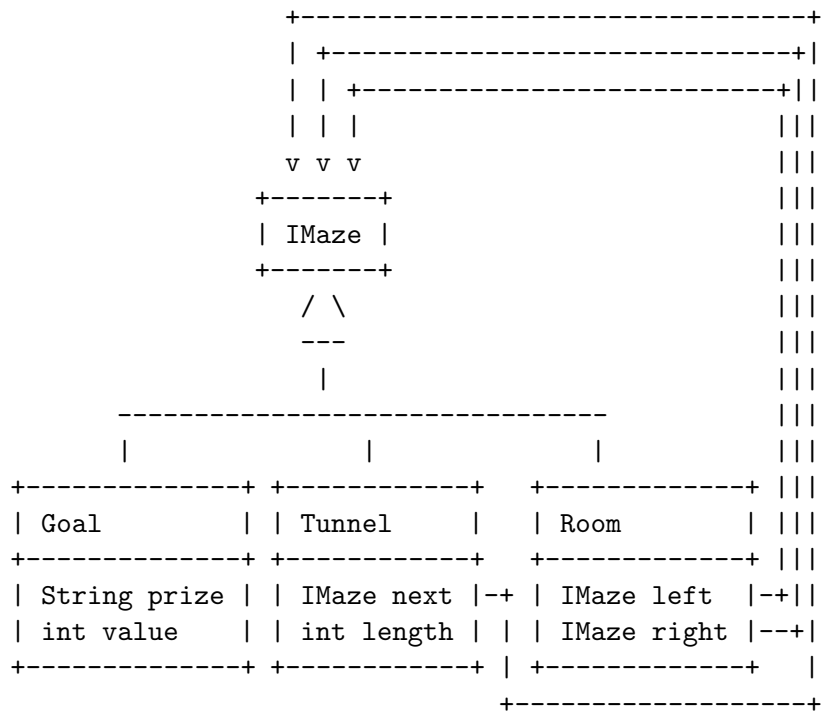
Name:

Student Id (last 4 digits):

• Write down the answers in the space provided.

• You may use all syntax that you know from *FunJava* other than *abstract classes*. 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 otherwise correct in FunJava.

• 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.

• We will not answer *any* questions during the exam.

Good luck.

Problem	Points	/
A		/ 0
B		/ 4
C		/10
D		/ 8
E		/ 4
Total		/26

Problem 1

Here is a Java class diagram that describes a maze with treasures to hunt
for:

```
                        +---------------------------------+
                        | +-------------------------------+|
                        | | +----------------------------+||
                        | | |                             |||
                        v v v                             |||
                     +-------+                            |||
                     | IMaze |                            |||
                     +-------+                            |||
                        / \                               |||
                        ---                               |||
                         |                                |||
         -------------------------------                  |||
         |                 |               |              |||
+-------------+ +-----------+   +-------------+ |||
| Goal        | | Tunnel    |   | Room        | |||
+-------------+ +-----------+   +-------------+ |||
| String prize | | IMaze next |-+ | IMaze left  |-+||
| int value    | | int length | | | IMaze right |--+|
+-------------+ +-----------+ | +-------------+   |
                             +------------------+
```
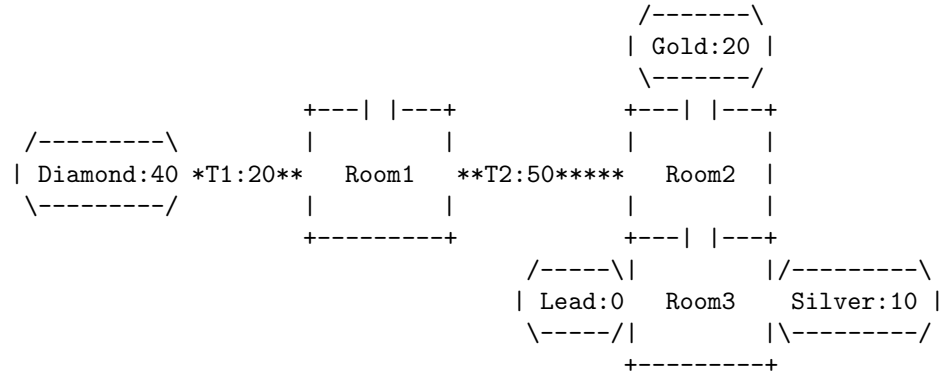
A. *(0 points)*

You are not required to do this. You may want to see what the data definitions look like. Do not spend much time on this, unless you do not understand what the data represents. The examples in the next part should make that clear.

Write down the Java class and interface definitions that are represented by this class diagram.

B. *(4 points)*

The following picture represents maze that can be represented by this data definition. Write down the data definitions needed to represent the information in this picture.

```
                                            /-------\
                                            | Gold:20 |
                                            \-------/
                        +---| |---+         +---| |---+
     /---------\        |         |         |         |
     | Diamond:40 *T1:20**   Room1   **T2:50*****   Room2  |
     \---------/        |         |         |         |
                        +---------+         +---| |---+
                                       /-----\|         |/---------\
                                       | Lead:0   Room3    Silver:10 |
                                       \-----/|         |\---------/
                                            +---------+
```

C. *(10 points)* Design the method `valueOf` that determines the value of the prize described by a given `String` in this maze. There may be several prizes with the same name, in that case add up their values.

D. *(8 points)*

Design the method `longestPath` that computes the longest path in a maze. The lengths of the tunnels are given (in feet), each room is 10 feet long and 10 feet wide.

... This page is intentionally left blank ...

E. *(4 points)*

Show the templates for all classes in this problem for which you have designed methods.