CSCI 151, Spring 2016 - Exam 1

Dr. Mark Goadrich

February 19, 2016

Show all of your work for full credit.

Name:

1. You are implementing the game TicTacToe. For each of the pieces listed below, say whether a *class*, *interface*, or *enum*, would be the most appropriate way to represent this in Java and why. You do not need to write any code for this question.

   (a) The board, which holds three rows and columns of spaces for play, and includes methods for placing marks on the board and determining when the game is over.

   (b) The contents of each space in the board, which can be an X, and O, or Empty.

   (c) A player of the game, which must include a method that returns a move given a board.

2. The following function contains errors. Identify each error and describe a repair.

```
public static String exclaim(String input) {
  String result = "";
  for (int i == 0; i < input.length(); i++) {
    if (input.substring(i, i+1) == ".") {
      result += ’!’;
    } else {
      return input;
    }
  return result;
}
```

Examples of expected results:

```
exclaim("Lemurs.") => "Lemurs!"
exclaim("Hello.  It’s me.") => "Hello!  It’s me!"
```

3. Describe an algorithm in pseudocode for how you would reverse the words in a sentence using a Stack.

   Examples of expected results:

   ```
   reverse("Sam I am") => "am I Sam"
   reverse("rain falls at bus stop") => "stop bus at falls rain"
   ```

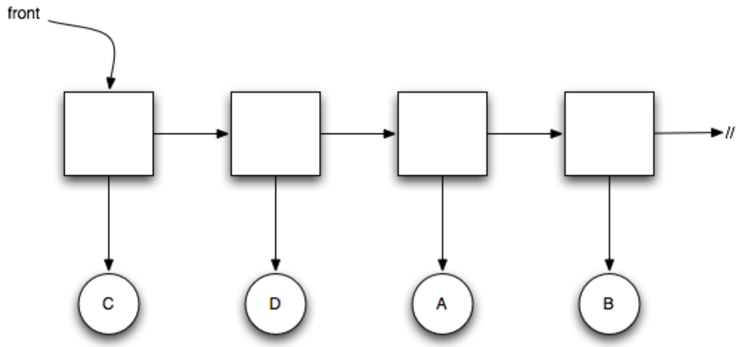
4. Popping from an ArrayIntStack can leave a large amount of extra space. Write a `shrink` method that will decrease the size of `stuff` by $\frac{1}{2}$ if the size of the stack is less than $\frac{1}{4}$ of the length of `stuff`.

```
public class ArrayIntStack implements IntStack {

   private int top;
   private int[] stuff;

   public ArrayIntStack() {
      top = -1;
      stuff = new int[8];
   }

   public int pop() {
      notEmpty();
      int temp = stuff[top];
      top--;
      shrink();
      return temp;
   }

   // COMPLETE THIS METHOD
   private void shrink() {









   }

   // OTHER METHODS NOT SHOWN
   ...
}
```

5. Complete the class below. The light should initially be off and the fan should be at speed 0. Valid speeds are 0, 1, 2 and 3.

```
public class CeilingFan {
   private boolean lightOn;
   private int speed;

   public CeilingFan() {


   }

   // Toggle the light off and on
   public void flipLight() {


   }

   // Decreases the speed by 1, or turns the fan to speed 3 if
   // the speed is currently at 0.
   public void pullChain() {



   }

   // Returns true if the light is on, otherwise false
   public boolean isLit() {


   }

   // Returns the current speed of the fan.
   public int getSpeed() {


   }
}
```

6. You are making a chain of nodes, and have access to the first in the chain using the identifier *front*, with the state given in the memory diagram below.



Write instructions in Java that will result in the below memory diagram, using only the *getNext* and *setNext* methods; you should not change the values of any nodes.