SUMMER 2007 CS1316: PRACTICE EXAM 2

A. DATA STUCTURES

a. Fill in the blank

i. It is easier to access a specific element in a specific location in a/an _____.

ii. It is easier to insert and delete in the middle of a/an _____.

iii. The fact that a/an _____is of a fixed length is both a pro and con.

b. Short Answer

i. What characterizes a static data structure? What is an example of a static data structure?

ii. What characterizes a dynamic data structure? What is an example of a dynamic data structure?

iii. Why is being of fixed length both a pro and a con?

B. STACKS and QUEUES

a. Short Answer

i. What is the difference between a Stack and a Queue?

ii. What are the names of the add and remove methods of a Stack?

iii. What are the names of the add and remove methods of a Queue?

iv. Describe in words how to add to a Queue.

v. Describe in words how to remove from a Queue.

vi. Describe in words how to add to a Stack.

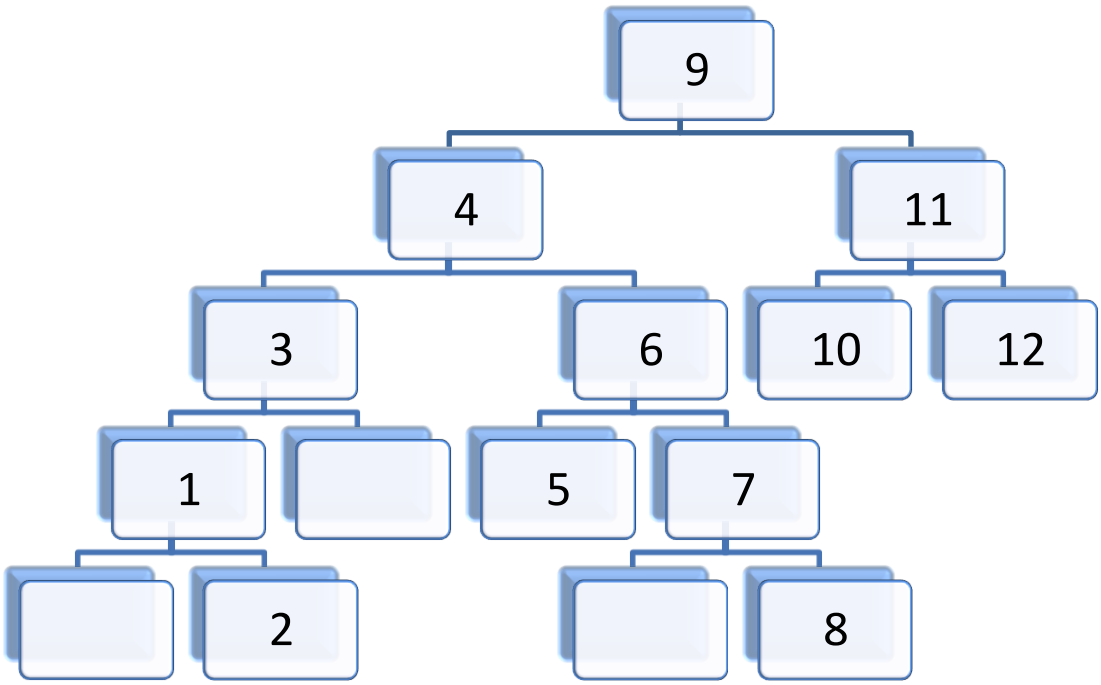
vii. Describe in words how to remove from a Stack.

C. TREES AND GRAPHS

a. What the difference between a Tree and a Graph?

b. What is so special about a binary tree?

c. TREE TRAVERSAL



i. Write the pre-order traversal of this tree.

ii. Write the in-order traversal of this tree.

iii. Write the post-order traversal of this tree.

d. GUI TREE

Consider the following code:

```
import javax.swing.*;
import java.awt.*;
public class ChatRoomGUI extends JFrame {
 String userName;
 public ChatRoomGUI(){
  super("Chat Room GUI");
  userName =”Colin”;
  this.setLayout(new BorderLayout());

  JPanel conversationPanel = new JPanel();
  JTextArea conversationBoxArea = new JTextArea("CS1316 Chat Room:", 10, 20);
  conversationBoxArea.setText(conversationBoxArea.getText()
    +"\nColin: So who is going to write the practice exam?");
  conversationBoxArea.setText(conversationBoxArea.getText()
    +"\nKristin: Not it!");
  conversationBoxArea.setText(conversationBoxArea.getText()
    +"\nRory: Not it!");
  conversationBoxArea.setText(conversationBoxArea.getText()
    +"\nDawn: Dang...");
  conversationPanel.add(conversationBoxArea);
  this.add(conversationPanel,BorderLayout.CENTER);

  JPanel listOfUsersPanel = new JPanel();
  JTextArea listOfUsersArea = new JTextArea("Colin\nDawn\nKristin\nRory");
  listOfUsersPanel.add(listOfUsersArea);
  this.add(listOfUsersPanel,BorderLayout.EAST);

  JPanel messageBoxPanel = new JPanel();
  JTextArea messageBox = new JTextArea(":(",5, 20);
  messageBoxPanel.add(messageBox);
  JButton sendButton = new JButton("Send");
  sendButton.addActionListener(
                        new ActionListener() {
    public void actionPerformed(ActionEvent e) {
      //see question below
    }});
  messageBoxPanel.add(sendButton);
  this.add(messageBoxPanel, BorderLayout.SOUTH);

  this.pack();
  this.setVisible(true);
 }
}
```

Draw the tree representation of the GUI tree structure. Refer to each component by its variable name and type such as panel1: JPanel.

What needs to be added inside the actionPerformed method in order for whatever was in the messageBox to appear in the conversationBoxArea (Assume that the new message will appear on a new line)?

D. SIMULATIONS

a. Short Answer

i. What is the definition of a simulation?

ii. What is a discrete simulation? What is an example of a discrete simulation?

iii. What is a continuous simulation? What is an example of continuous simulation?

iv. There is a 40% chance that Robbie, a rookie soccer player, will kick the ball into the goal. We can represent this probability by using Math.random():

```
if(Math.random() < .40)
  scoresl();
else
  misses();
```

1. How would we represent this probability using the Random class?

2. How would we represent this probability using Math.random() if Robbie practiced really hard and now has a 25% chance of missing the goal.

v. TRY-CATCH BLOCKS AND FILE I/O

Consider the following code:

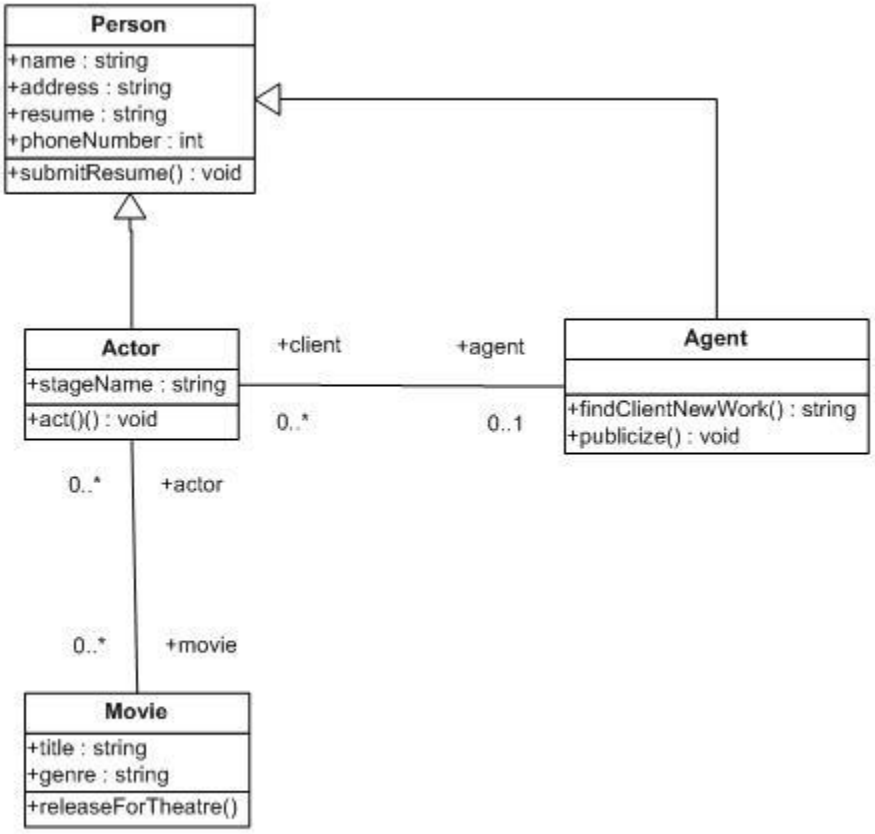
```
import java.io.*;

try{
      BufferedWriter output = new BufferedWriter(
        new FileWriter("C:/practiceExam.txt"));
      output.write("CS1316 rocks my socks! ");
      output.newLine();
      output.write("However, I am nervous about this exam.");
      output.close();
    } catch (Exception ex) {
      System.out.println("Could not write the data!");
      System.out.println(ex.getMessage());
    }
```

1. What will happen after this code is executed? What will be inside the text file?

2. Why is a try-catch block necessary?

E. UML DIAGRAM



a. Which classes have the instance variable name?

b. Which classes understand the method act()?

c. How many Actors can be associated with a given Movie?

d. How many Agents can an Actor have?

e. Which classes have the instance variable stageName?

f. Which classes understand the method submitResume()?

F. STRUCTURING IMAGES AND LINKEDLIST METHODS

Implement the following method in PositionedSceneElement:

public void findAndReplaceRepeat(PositionedSceneElement oldelement, PositionedSceneElement newelement, int n)

Find oldelement, remove it from the list and then insert at n copies of newelement in its place. Handle the case where oldelement is never found.

public void findAndReplaceRepeat(PositionedSceneElement oldelement, PositionedSceneElement newelement, int n)

G. MUSIC AND SONGNODES

Use the following code to assist you with the following problem:

```
public void repeatNextInserting(SongNode nextOne, int count){
  SongNode current = this;
  for (int i=1; i <= count; i++){
    SongNode copy = nextOne.copyNode();
    current.insertAfter(copy);
    current = copy;
  }
}

public void repeatNext(SongNode nextOne,int count) {
  SongNode current = this;
  SongNode copy;
  for (int i=1; i <= count; i++){
    copy = nextOne.copyNode();
    current.setNext(copy);
    current = copy;
  }
}

public void weave(SongNode nextOne, int count, int skipAmount){
  SongNode current = this;
  SongNode copy, oldNext;
  int skipped;
  for (int i=1; i <= count; i++){
    copy = nextOne.copyNode();
    skipped = 1;
    while ((current.next() != null) && (skipped < skipAmount)){
      current = current.next();
      skipped++;
    }
    oldNext = current.next();
    current.insertAfter(copy);
    current = oldNext;
    if (current == null)
      break;
  }
}

public void insertAfter(SongNode nextOne){
  SongNode oldNext = this.next();
  this.setNext(nextOne);
  nextOne.setNext(oldNext);
}
```

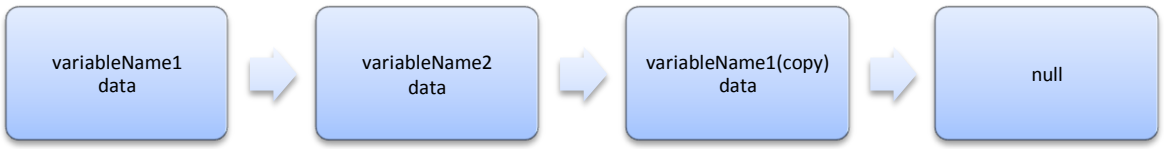
a. What would be the resulting list after the following lines of code is executed?

```
SongNode node1= new SongNode();   node1.setPhrase(SongPhrase.AG1());
SongNode node2= new SongNode();   node2.setPhrase(SongPhrase.AG2());
SongNode node3= new SongNode();   node3.setPhrase(SongPhrase.riff1());
SongNode node4= new SongNode();   node4.setPhrase(SongPhrase.riff2());

node1.insertAfter(node2);
node2.insertAfter(node3);
node3.insertAfter(node4);

node1.repeatNextInserting(node2, 2);
node3.repeatNext(node4,3);
node1.weave(node3, 3, 1);
```

When answering please include the node's reference name (variable name) and its data. If the node is a copy of another node, its reference name is the name of the node it was copied from plus a (copy). Also please format answers like so:



b. How come we can call the AG1() method from SongPhrase without creating an instance?