

I commit to uphold the ideals of honor and integrity by refusing to betray the trust bestowed upon me as a member of the Georgia Tech community.

CS 1316 Exam 3 Fall 2009

Section/Problem	Points Earned	Points Possible
1. Terms & Concepts		33
2. Short Answers		12
3. Trees		9
4. Tree Question		5
5. Queue Operations		3
6. Code Comprehension GUI		10
7. Write Code: Musical Instrument		15
Total Points:		87

1. Terms & Concepts (33 points)

For each of the terms below, write 1 or 2 sentences defining the term and proving you understand what your definition means. You may include an example if you think it will help your explanation. Be concise and precise.

1. abstract (super) class

2. anonymous inner class

3. binary search tree

4. final static variable

5. interface

6. layout manager

7. leaf node

8. queue

9. static method

10.this (keyword)

11.void (keyword)

2. Short Answer (12 points)

a. What causes a null pointer exception? (2pts)

b. What is the difference between a compiler error and a runtime exception? Give an example of each. (3pts)

c. Multiple Choice: Interfaces contain the following: (check all that apply) (1pt)

method signatures

method bodies

static, final fields

instance variables

d. What is the main difference between a continuous and a discrete event simulation? (3pts)

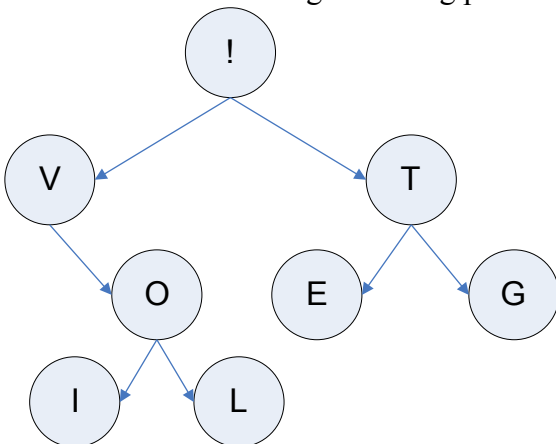
e. How is sampled sound stored in a computer? What does each “sample” represent? (3pts)

3. Trees (9 points)

a. If the in-order traversal of the balanced binary search tree T is: M N O P Q R S, draw the tree:

b. If the pre-order traversal of the binary search tree T is: F E D C B A, draw the tree:

c. Traverse the following tree using post-order traversal. Write out each letter.



4. Tree Question (5 points)

Consider the method X:

```
public static int X( TreeNode t) {
    if (t.left == null && t.right == null) {
        return t.data;
    } else if (t.left == null) {
        return Math.min( t.data, X(t.right) );
    } else if (t.right == null) {
        return Math.min( t.data, X(t.left) );
    } else {
        return Math.min( Math.min( t.data, X(t.left) ), X(t.right) );
    }
} // end X
```

If you call X giving it a non-empty tree, what statement can you make about the value that X returns?

5. Queue operations (3 Points)

Examine the following series of code statements. Draw the contents of the queue after the following code executes. Clearly mark the front and back of the queue.

```
Queue x = new Queue();
x.enqueue("A");
x.enqueue("B");
x.enqueue("C");
String temp1 = x.dequeue();
String temp2 = x.dequeue();
x.enqueue("D");
```

6. Code Comprehension - GUI's (10 points)

Draw the GUI that will be made when the following code is run. If no text would be shown on the widget when the GUI is displayed, write the name of the type of widget instead (button, slider, etc)

```
import java.awt.BorderLayout;
import java.awt.GridLayout;
import javax.swing.*;
public class Test3GUI extends JFrame{
    public Test3GUI() {
        super("Test3GUI");
        this.getContentPane().setLayout(new BorderLayout());
        JLabel label1 = new JLabel("Label1");
        JTextField text = new JTextField(20);
        JButton b1 = new JButton("Button1");
        JButton b2 = new JButton("Button2");
        JButton b3 = new JButton("Button3");
        JSlider slider = new JSlider();
        JComboBox cb = new JComboBox();
        JPanel p1 = new JPanel();
        p1.setLayout(new GridLayout(2,1));
        this.add(label1, BorderLayout.NORTH);
        p1.add(b1);
        p1.add(b2);
        this.add(p1, BorderLayout.EAST);
        this.add(cb, BorderLayout.WEST);
        this.add(slider, BorderLayout.SOUTH);
        this.add(text, BorderLayout.CENTER);
        this.add(b3, BorderLayout.CENTER);
        this.pack();
        this.setVisible(true);
    }
    public static void main(String[] args) {
        Test3GUI qui = new Test3GUI();
    }
} // end class
```

7. Create an Object: Musical Instrument (15 points)

You are given the following interface:

```
public interface MusicalInstrument {  
  
    // returns the sound of the instrument  
    public Sound getSound();  
  
    //returns the name of the musical instrument  
    public String getName();  
  
}
```

Your task is to create a new class called Piano that fully implements the MusicalInstrument interface. Your new class should also contain the following:

- A public instance variable of type Sound
- A constructor that accepts a single parameter of type Sound and sets the sound instance variable to the input sound.
- An additional method called appendNPlay that takes in a sound “A” and a double “factor”, and does not return anything. This method should append the input sound to the instance variable sound. After it does that, it must also modify the volume of the resulting sound by the factor parameter. (Multiply each sound sample by the factor, and put the new value back into the sample.) Once you have completed your manipulations, play the resulting sound!

Methods in the Sound class that you might find useful:

- Sound append (Sound appendSound)
- void play()
- SoundSample [] getSamples()

Methods in the SoundSample class that you may find useful:

- int getValue()
- void setValue(int)

Write your code on the back of this page:

Your Name: _____