

CS 1316 - Exam 3 - Spring 2010

Name: _____

Grading TA: _____ Section: _____

INTEGRITY: By taking this exam, you pledge that this is your work and you have neither given nor received inappropriate help during the taking of this exam in compliance with the Academic Honor Code of Georgia Tech. Do NOT sign nor take this exam if you do not agree with the honor code.

DEVICES: If your cell phone, pager, PDA, beeper, iPod, or similar item goes off during the exam, you will lose 10 points on this exam. Turn all such devices off and put them away now. You cannot have them on your desk.

ACADEMIC MISCONDUCT will not be tolerated. You are to uphold the honor and integrity bestowed upon you by the Georgia Institute of Technology. Penalties for misconduct will be a zero on this exam, an F grade in the course, and/or other disciplinary action.

- Keep your eyes on your own paper.
- Do your best to prevent anyone else from seeing your work.
- Do NOT communicate with anyone other than a proctor for ANY reason in ANY language in ANY manner.
- Do NOT share ANYTHING during the exam. (This includes no sharing of pencils, paper, erasers).
- Follow directions given by the proctor(s).
- Stop all writing when told to stop. Failure to stop writing on this exam when told to do so is academic misconduct.
- Do not use notes, books, calculators, etc during the exam.

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.

Signature: _____

Problem	Points Earned	Points Possible	Grader Initials
1. Vocabulary		12	
2. Tree Travesals		6	
3. GUI Code Reading		6	
4. Code Reading		10	
5. Try/Catch		4	
6. Small Questions		4	
7. Linked List		10	
TOTAL:		52	

Exam Percentage: _____ / 52 = _____ %

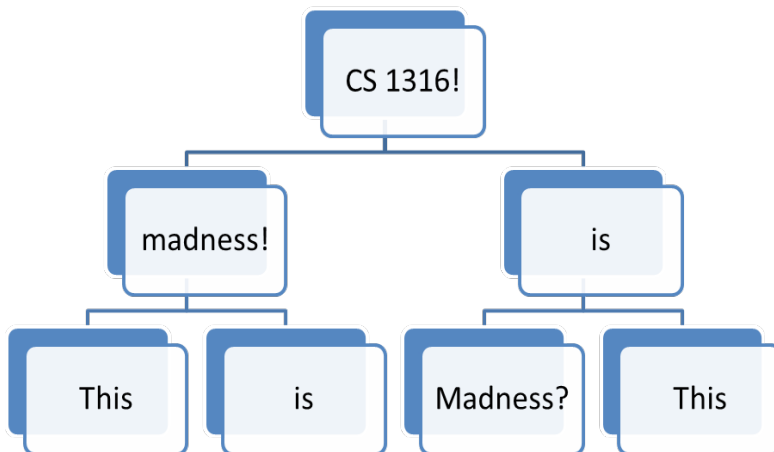
1. Vocabulary (12 points)

For each of the following words, write a 1-2 sentence definition of the word as used in this class. Your definition should be concise and to the point, while demonstrating that you know what the term means.

- a) binary search tree -
- b) recursive traversal (of a list) -
- c) simulation -
- d) user interface events -

2. Tree Traversals (6 points)

Examine the following tree. Write the correct strings stored in the nodes for each of the following traversal methods.



Pre-Order:

In-Order:

Post-Order:

3. GUI Code Reading (6 points)

You are given the following lines of code:

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;

public class MysteryGUI extends JFrame implements ActionListener{
    JButton jb;
    JLabel jl;

    public MysteryGUI() {
        this.getContentPane().setLayout(new BorderLayout());

        JPanel panell = new JPanel();
        jb = new JButton("Click me!");
        panell.add(jb);
        this.getContentPane().add(panell, BorderLayout.NORTH);
        jb.addActionListener(this);

        JPanel panel2 = new JPanel();
        jl = new JLabel("Hello");
        panel2.add(jl);
        this.getContentPane().add(panel2, BorderLayout.SOUTH);

        pack();
        setVisible(true);
    }

    public void actionPerformed (ActionEvent e) {

        if (e.getSource() == jb) {
            jl.setText ("Good bye");
        }
    }
}
```

1. Draw the GUI the code above produces initially (4 points):

2. What happens when we click the JButton jb (2 points)?

4. Code Reading(10 points)

Examine the code below. For each commented position (1-6) determine whether the access operation is legal or not. If it is illegal, write "ERROR" and briefly explain why it is illegal. If it is legal, write "LEGAL". In determining your answers, ONLY consider the position in question; that is, for each part, *assume that all of the other code works correctly, even if you believe it would not in reality.*

```
public class Landlord{
    private String name;
    public Building building;
    private static double income = 0.0;

    public Landlord(String name, Building building){
        this.name = name;
        this.building = building;
    } //end constructor

    public int headCount(){
        int count = building.numTenants; // _____ Position 1
        return count;
    } //end headCount method

    private double getIncome(double rate){
        return building.income(rate); // _____ Position 2
    } //end getIncome method

    public static void main(String[] args){
        Building b = new Building("Woodruff", null);
        Landlord bob = new Landlord("Bob", b);
        double income = getIncome(500.0); // _____ Position 3
        if(income < 50000){
            Building.demolish(); // _____ Position 4
        }
        System.out.println("Income is " + bob.getIncome(500.0));
    } //end main method
} //end Landlord class

public class Building{
    private String address;
    private static int numTenants = 0;

    public Building(String a, Landlord landlord){
        address = a;
        landlord.building = this; // _____ Position 5
    } //end constructor

    public static void demolish(){
        address = "Building no longer exists"; // _____ Position 6
    } //end demolish method

    private double income(double rate){
        return numTenants * rate;
    } // end income method
} //end Building class
```

5. Try/Catch (4 points)

Examine the following code:

```
try {  
    Picture p = null;  
    p.getPixels();  
    System.out.println("Open the pod bay doors, HAL.");  
}  
catch (Exception NullPointerException) {  
    System.out.println("I'm sorry, Dave.");  
    System.out.println("I'm afraid I can't do that.");  
}
```

- a. Circle the line that would throw an exception.
- b. Write the text that the code would output to the interactions pane:

6. Small Questions (4 pts)

FIFO is an acronym for:

- a) First In Final Out
- b) Final In First Out
- c) First In First Out
- d) Final In Final Out

Suppose there is a binary tree T with 14 nodes. What is the minimum possible height (depth) of T?

- a) 0 b) 3 c) 4 d) 5 e) 1 f) 2

Draw a binary search tree where the in-order traversal of nodes produces: 6, 8, 10, 12, 16, 18

7. Linked List (10 points)

Write a static method named `goTo` that takes in an integer location and a pointer to an `LLNode`. It should return the (String) data in the appropriate node. Assume that all node contents/data are strings and that the `LLNode` class has a method called `getData()` that returns the String within a node. Also assume that `LLNodes` have a `getNext()` method that returns a reference to the next node in the list (or null if you are at the end of the list). You should start counting the elements in your list at 0. You **must check for invalid inputs**--negative inputs or inputs beyond the length of the list. If the inputs are not valid return null **and** print an appropriate message to the interactions pane.

Ex: if you have the following list and you call `goTo(2,head)` it should return "Test3"

