1. Java chose to utilize single inheritance instead of multiple inheritance like C++. Explain the problem with multiple inheritance and provide a code snippet to support the problem. (1%)

2. Many programmers believe the garbage collector in Java will not allow for memory leaks. Explain how a memory leak could still occur in Java despite having a garbage collector. (1%)
3. Convert the following C++ code to Java (2%): 

```cpp
#include <iostream>
using namespace std;
class S {
    protected:
    int n;
    public:
    S(int n) {
        this->n = n;
    }
    virtual int g() = 0;
};
class T : public S {
    private:
    int n;
    public:
    T(int m) : S(m) {
        n = ++m;
    }
    int g() {
        return n;
    }
    int h() {
        return S::n;
    }
};
void p(S &s) {
    cout << s.g() << endl;
}
void main() {
    T t(5);
    cout << t.g() << endl;
    cout << t.h() << endl;
    p(t);
}
```
4. Lines of code that can throw unchecked exceptions do not need to be inside of a try block. Explain why this is the case. (1%)

5. Give two reasons why a programmer would choose to use an anonymous inner class rather than a non-anonymous class? (0.5% + 0.5%)

6. Which of the following acknowledge the preferred height and width of a component? Circle all that apply. (1%)

   BorderLayout  FlowLayout  CardLayout
   GridLayout    GridBagLayout BoxLayout
   GroupLayout    null layout
7. Provide the code to setup the following JFrame. Include all of the layout managers, panels, and components. Do not include any actions. Do not use the null layout. (2%)
9. When using JTrees, in what class is the data for the tree held? (1%)

10. What is the advantage to using generics instead of just overriding methods? (1%)

11. Draw the GUI generated by the following code. (1%)
Object[] options = {"Approve", "Deny"};
int value = JOptionPane.showOptionDialog(MyFrame.this, "Terms and Agreements",
  "Licensing of Application",
  JOptionPane.OK_CANCEL_OPTION,
  JOptionPane.WARNING_MESSAGE,
  null,
  options,
  options[0]);