Question 1 (10 pts)

a) How does the intent of the Adapter pattern differ from the Façade pattern? (2pts)

b) What is the difference between Reliability and Availability of a software system? (2pts)

c) How are the Strategy and State design patterns i) similar? ii) different? (2pts)

d) What is the Open Close design principle? (2pts)

e) Would using a metric such as LOC (Lines of Code) help you with assessing the psychological complexity of a program? (2pts)
Question 2 (10 pts)

Answer True or False.

a) Branch coverage and statement coverage are not equally powerful.

b) If we achieve 50% branch coverage then we satisfy the requirement to have perfect testing.

c) Flowgraphs can be used in object oriented testing.

d) The state of an object is determined ONLY by the values stored in the object’s attributes at a given point in time.

e) Mutation testing helps us evaluate the adequacy of test data.

f) All test scenarios in a decision table must be mutually exclusive.

g) All test scenarios in a decision tree must be mutually exclusive.

h) Failures are a consequence of software bugs.

i) The Yoyo problem exemplifies problems with testing object oriented inheritance hierarchies.

j) A testing oracle is a correct specification of software.
Question 3 (20 pts)

a) Draw a UML activity diagram to process an order (i.e. buying a book from Amazon). Your diagram should have activities that represent the processing of the order, the shipping of the order, and canceling the order. An order may be cancelled whilst in the processing stage. (5pts)

b) Draw a UML activity diagram that depicts how a search engine returns results. This diagram should have a fork that allows the engine to improve its results. The window for improving the results is only 3 seconds. When the 3 second timeout occurs, the search results are returned and the entire activity ends, including the improving the search results activity. However, if the improve the search results activity finishes before the 3 second timeout, it will not stop the overall activity. (5pts)
c) Draw a picture that contrasts the process of refactoring code when an anti-pattern is found and that of using a design pattern to solve a problem. (5pts)
d) Draw a control flowgraph for the following code (5pts):

```java
public int displayMsg(int nToPrint)
{
    np = 0;
    if ( (msgCounter > 0) && (nToPrint > 0) ) {
        for (int j = lastMsg; (j != 0) && (np < nToPrint); --j) {
            System.out.println(messageBuffer[j]);
            ++np;
        }
    } else if (np < nToPrint) {
        for (int j = SIZE; (j != 0) && (np < nToPrint); --j) {
            System.out.println(messageBuffer[j]);
            ++np;
        }
    }
    return np;
}
```
Question 4 (20 pts)

We are all familiar with IDEs. (Integrated Development Environments). Choose your favorite IDE and provide the following:

a) **A high level component architecture.** Use a UML component diagram to clearly show the different components of your architecture. After you have identified your components, show how they interact with each other by clearly showing the *provided/needed* interfaces, *ports, connectors*, etc. (10 pts)

b) Choose one component from your architecture (described in part a), and draw the **detailed UML class diagram.** For example, if your IDE of choice is Eclipse, then a possible component may be one of the plug-ins (i.e. SVN, CVS, PlantUML, database tools, etc.) (10 pts)
Additional Space