1. a. Describe briefly the distinction between black-box testing and white-box testing of a program module (10 points)

See http://webpages.cs.luc.edu/~cweisert/COMP370/foils/Session4a.pdf, #5

b. Describe (or give an example of) one situation in which either is likely to reveal an error that might be overlooked by the other and explain why: (10 points)

Boundary conditions on branches (if statements) and limits on iteration may be:
- inverted
- off by one
- redundant
- or otherwise wrong

Black-box testing wouldn’t know about those critical boundary values, but white-box would focus on them.

On the other hand (but less likely) white-box testing might make invalid implementation-based assumptions, duplicating the programmer’s own earlier oversights.

2. You’ve been assigned to design and develop a Person composite class. Identify two elementary-item classes that you’ll probably also need for member data items in that class. (5 points each)

PersonName, Sex (or Gender), Date, Identifier, MaritalStatus,

Note: We asked for classes not individual data items. DateOfBirth and DateHired are instances of Date.

3. For many data items we distinguish between an appropriate external representation and an appropriate internal representation. Explain why, and state the most important criteria for choosing each. (13 points)

External representations should be:
- familiar to people who use the application
- not error prone

Internal representation should be:
- simple
- efficient
- flexible, expandable
4. Suppose a program module produces correct results for every test case. What other criteria should we consider in assessing the quality of that program module? (12 points)

- Readability, ease of understanding.
- Modularity, ease of diagnosing problems, implementing changes.
- Efficiency (time and space).

5. In developing a major new application, we perform systems analysis and system design. Explain the difference between the two. (13 points)

The end-product of Systems Analysis is a specification defining exactly what the proposed application will do. It is usually produced by systems analysts in consultation with prospective users and user-management.

A System Design is usually produced by technical specialists (programmers, database designers, etc.) It defines how the new application system will be built: databases, program modules, object-oriented classes, etc. It must, of course, be a faithful implementation of the system specification produced by the analysts.

In our sample life cycle, SA was done mainly in phase 3, SD mainly in phase 4.
6. For each of the following program fragments, explain what, if anything, is seriously wrong with it. If nothing is seriously wrong, just leave it. (8 points each)

a. public Date operator+(Date d1, Date d2)

Adding two dates is meaningless. (You can add a date and a duration yielding a date, or you can subtract one date from another date yielding a duration.)

b. public void arraySum(double [] dta)
{
double total = 0.0;
for (int k=0; k < dta.length; ++k)
    total += dta[k];
return;
}

The result is thrown away! Change void to double and return the result.

Note: There’s nothing wrong with using a return statement with void function. In fact, for clarity it’s preferred to just falling off the end. But here no result is available to the calling program.

A couple of students thought that ++k should be replaced by k++. In this context, however, (not inside a larger expression) they produce the same result. Actually the prefix form is preferred, as explained in http://www.idinews.com/peeves/prefixIncr.html.

Because we didn’t discuss those conventions in class no penalty was assessed for citing either of them. But you were penalized if you didn’t notice that the result of the function was thrown away.

c. public struct Temperature {
    double value;
    char units; // F: value is in Fahrenheit degrees
    // C: value is in Celsius degrees
    .
}


d. public class Weight {
    double pounds;
    int ounces;
    .

See http://www.idinews.com/mixedBase.html