

ICS52 - Introduction To Software Engineering
Final Exam – Fall, 2003

First Name: _____ Last Name: _____

Student ID: _____

1. (24 points, 4 points each) Define the following terms, as used in software engineering:

A. Test Case.

**A function/method to test, an input, and an expected output.
(2 pts for just “input”)**

B. Verification.

Are we building the product right? Making sure the software conforms to its specifications.

C. The repository model.

**A software architecture model in which all data is kept in one module,
and that module communicates the data to other modules.**

D. Stub.

A small routine which substitutes for a larger routine which is not available.

E. Architectural design.

Identifying sub-systems and their control and communication.

F. Configuration Management.

**The development and application of standards and procedures for managing
an evolving system product.**

2. (16 points) You have been assigned to design test cases for black box testing of the two parameter substring method in java.lang.String. From the documentation:

```
public String substring(int beginIndex, int endIndex)
Returns a new string that is a substring of this string. The substring begins at the
specified beginIndex and extends to the character at index endIndex - 1. The
length of the substring is endIndex-beginIndex. Throws
IndexOutOfBoundsException if the beginIndex is negative, or endIndex is
larger than the length of this String object, or beginIndex is larger than
endIndex.
```

4 pts for each part

- (a) What is the input domain of the substring function?

A String and two ints. (-2 for just "two ints")

- (b) What is a basis for dividing the input domain you described into subdomains?

Sample correct answer: The size of endIndex – beginIndex (assuming this size is < the String's length).

-1 at least if your answer is based on the *output* not the input

- (c) Using the basis from (b), name three or four subdomains.

Size = 0, 1, 2-3, 4 or more.

- (d) For each subdomain from (c), give a test case input and the expected output.

For all test cases, the String is "Look on my Works, ye mighty, and despair!"

	Subdomain: 0, 1, 2-3, 4+	Expected output
begin=5 end=5	x	""
begin=11 end=12	x	"W"
begin=8 end=10	x	"my"
begin=21 end=27	x	"mighty"

-2 if exact values not specified

3. (12 points) Describe in a few words each quadrant of the spiral model.

3 points each, see text and notes for answers

A. Upper left (northwest).

B. Upper right (northeast).

C. Lower right (southeast).

D. Lower left (southwest).

4. (10 points) Select one of the software process models described in the textbook or in lecture, other than the spiral model, and briefly describe how the development of the ATAMS system, as described in the Scientific American article, followed the approach of the process model you selected.

Lots of possible correct answers, of course. Points off if you did not describe a fairly natural fit between the ATAMS development and the model. A common answer that didn't get too many points was "Waterfall model because the waterfall model has several steps and the ATAMS development had several steps." An answer like that doesn't capture the essence of either the model or the development process.

5. (10 points) According to Dijkstra, "Program testing can be used to show the presence of bugs, but never to show their absence." Is Dijkstra's dictum true if the program testing achieves node coverage? Explain why or why not.

Yes it is true, but we gave some partial credit to "no" answers that were reasonably well-reasoned.

6. (6 points) Maintenance costs for a system are typically higher than the original development costs. The textbook lists several "key factors that distinguish development and maintenance and which lead to higher maintenance costs." Name and briefly describe one of these factors.

See p. 608-609.

Team stability – after delivery, the development team is often broken up

Contractual responsibility – the development team's contract does not provide an incentive to make the software easy to maintain

Staff skills – maintenance staff are often inexperienced, unfamiliar with the application domain

Obsolete programming language – old systems may be written in obsolete programming languages that the maintenance staff has to learn

Program age and structure – an aged system may have a structure that is hard to understand, due to the many changes it has undergone.

7. (10 points) Below are two alternate versions of part of a School module's design. In each case, the **getStudentsSortedByGradeAndName** method is supposed to return information about the Students in the School, in a specified range of grades, and in alphabetical order by the name of the student.

```
// version 1
class School
{
    Enumeration getStudentsSortedByGradeAndName
        (GradeLevel start, GradeLevel end);
}

// version 2
class School
{
    Student[] getStudentsSortedByGradeAndName
        (GradeLevel start, GradeLevel end);
}
```

Discuss the relative benefits and shortcomings (if any) of each design approach, referring to specific software qualities and principles discussed in lecture or the textbook.

The weakness of version 2 is that it either exposes an internal data structure (the sorted array), or it is inefficient, because an array will have to be constructed.

Version 1 shows greater qualities of information hiding and abstraction.

The rubric was 4 points for the first two benefits/shortcomings (if they were correct and well-stated) and 2 points for the third, so at least three needed to be mentioned for full credit.

8. (3 points) Black box testing is another name for (choose the best answer):
- A. **Specification-based testing.**
 - B. Structural testing.
 - C. Verification.
 - D. Unit testing.
 - E. Stress testing.
9. (3 points) Which of the following is a reason why the complete path coverage condition is almost impossible to achieve? (Choose one.)
- A. **The number of execution paths is very large.**
 - B. The number of nodes in the control flow graph is very large.
 - C. Each test case can traverse at most n paths, where n is the number of nodes in the control flow graph.
 - D. Some loops in the program would have to be skipped.
 - E. Test cases can only test reachable statements.
10. (3 points) Checking that changes made to a program have not introduced new errors is called (choose one)
- A. **Regression testing. P. 423**
 - B. Verification.
 - C. Cleanroom inspection.
 - D. Validation.
 - E. Static analysis.
11. (3 points) Which of the following is *not* true of integration testing?
- A. **It should be performed before individual programs are tested. p. 452**
 - B. It determines whether modules make compatible assumptions about each other.
 - C. It can be done in a top-down or bottom-up fashion.
 - D. It often reveals errors due to invalid assumptions about interfaces.
 - E. It typically requires additional software called drivers.