

DECEMBER 2004 EXAMINATIONS

CSC 408H1F/CSC 2105H1F
Software Engineering

Duration: 2 hours.
One 8 ½ by 11 aid sheet is permitted.

Student number _____

Name _____
(surname) (given name)

INSTRUCTIONS:

1. Print your name and student number on this examination paper.
2. Check this examination paper. There are 7 pages including this cover page.
3. All questions are to be answered directly in this examination paper.
4. The mark for each question is given in the right-hand column: [points x weights], and the total number of marks is 100. This test is worth 35% of your final grade.

HINTS:

1. Read each question carefully.
2. Use a pen, not a pencil.
3. Write legibly and understandably. An unreadable answer receives no credit.
4. You don't have to write in full sentences. Phrases or points will suffice.
5. For each question, spend time appropriate to the number of marks given.
6. If there seems to be ambiguity in the question, ask the invigilator for clarification or write your assumptions and continue. If your assumptions are reasonable and don't trivialize the problem, you will probably receive full credit.

Don't write in the table below:

Topic	Questions	Marks available	Marks received
Principles	1-5	27	
Process and evolution	6-10	18	
Requirements and testing	11-13	22	
Management issues	14-17	18	
Software reuse	18-21	15	
	Total	100	

PRINCIPLES

1. What is the central theme of Software Engineering? [1 x 3]
Central Theme: _____

2. What are the ideal goals of Software Engineering? [4 x 2]
Goal 1: _____
Goal 2: _____
Goal 3: _____
Goal 4: _____

3. According to Lehman, there are 5 laws of software evolution. Can you remember 3 of them? [3 x 1]
LAW 1: _____
LAW 2: _____
LAW 3: _____

4. Identify 3 good software design principles. How can you apply them to the *OmniEditor*? [3 x 2]
Principle 1: _____
Web service design: _____

Principle 2: _____
Web service design: _____

Principle 3: _____
Web service design: _____

5. Indicate whether the following statements are true or false. For two of your choices, you may provide an explanation. [6 x 1 +2 x 1]
T / F

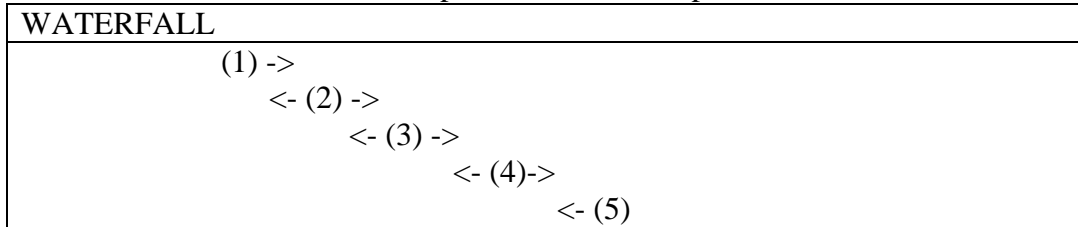
- _____ 1. Large-scale software development can only be done by a large group of software engineers.
- _____ 2. Interoperability and reusability are among the most critical issues in large-scale software development.
- _____ 3. Software documentation leads to the success of a software project.
- _____ 4. Delayed software has a bad product quality.
- _____ 5. Fixing an error in requirements costs much less than fixing a customer defect.
- _____ 6. Open source software can reduce cost to 0.

Explanation 1. _____

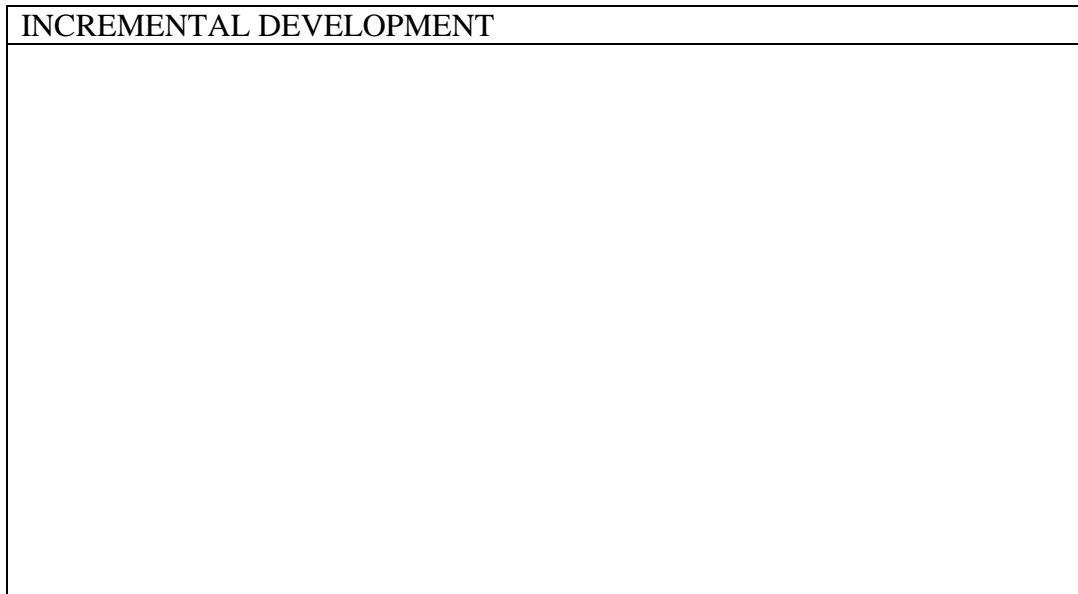
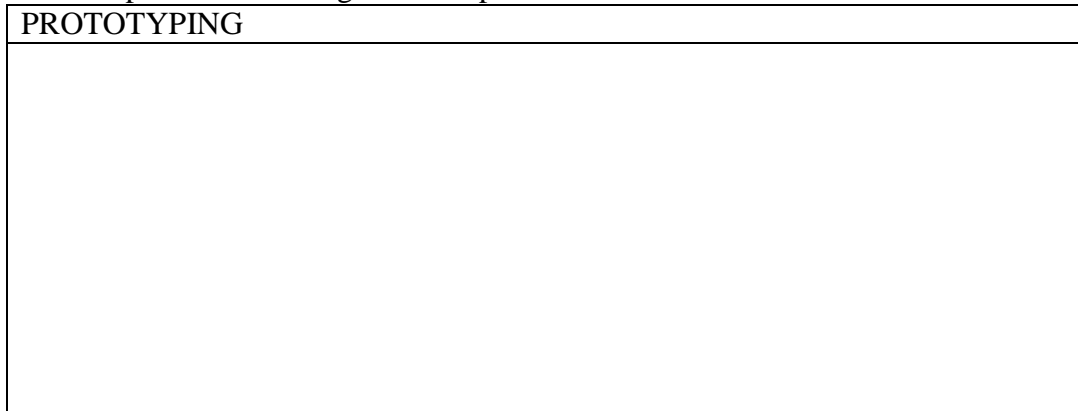
Explanation 2. _____

PROCESS AND EVOLUTION

6. The major phases in a software development project are: (1) requirements engineering (2) design (3) implementation (4) testing (5) maintenance. A water fall process model is depicted as follows. [2 x 3]



Please depict the following software processes:



7. What is the 40-20-40 rule for software development process? [1 x 3]

8. What is the difference between verification and validation? [1 x 2]
 Verification is: _____

Validation is: _____

9. Extreme Programming is a practice that two programmers frequently switch the role of coding and testing. When does it help productivity? [2 x 1]
 When does it hurt productivity?

HELP PRODUCTIVITY _____

HURT PRODUCTIVITY _____

10. What is the critical path in a project plan? _____ [2 x 2]

How is it related to improving productivity? _____

REQUIREMENTS SPECIFICATION AND TESTING

11. Please specify the requirements for the “upload” and “download” functions in the *OmniEditor* Web Service: [12 x 1]

	Upload	Download
Goal		
Input		
Precondition		
Output		
Post-condition		
Exception		

12. Please give 3 test cases for the “upload” and “download” functions you defined in Question 11. [6 x 1]

	Upload	Download
Test 1		
Test 2		
Test 3		

13. What is the difference between a fault and a failure? [4 x 1]

A fault is: _____

A failure is: _____

How are they related to correctness and reliability? A fault is related to _____, a failure is related to _____.

PROJECT MANAGEMENT

14. The Capability Maturity Model (CMM) has 5 levels: (1) initial, (2) repeatable, (3) defined, (4) managed and (5) optimizing. Please identify your *OmniEditor* team’s CMM level and give reason why it is at that level. We are at CMM level _____. The reason is _____ [2 x 1]

15. A company in India or China has the optimizing quality management. Will you outsource your project to them? Give a reason why (why not) [2 x 1]

16. Consistency recompilation is a configuration management problem. To solve it, the configuration items need to be recompiled should be MINIMIZED whenever a configuration item is changed. [3 x 3]

In C/C++ development, a compilation unit “.c” file corresponds to a module, and an include header unit “.h” file corresponds to an interface, see following example.

```

/* foobar.h */                /* moo.h */
void foo();                    void moo();
void bar();

-----
/* foo.c */                    /* bar.c */
#include "foobar.h"            #include "moo.h"
void foo() {                  void bar() {
    bar();                    moo();
}                              }

```

T / F

- _____ 1. One module needs to be recompiled if the interface it provides has changed
- _____ 2. One module needs to be recompiled if the interface it uses either directly or indirectly has changed
- _____ 3. One module needs to be recompiled if the modules providing the interface has changed

Indicate whether the above statements are true or false. Support each of your argument using the above-listed code.

- 1. _____
- _____
- 2. _____
- _____
- 3. _____
- _____

Give an example using *OmniEditor*, consider the WSDL as an interface between the web service modules and the client modules.

- 1. _____
- _____
- 2. _____
- _____
- 3. _____
- _____

17. List 4 kinds of software maintenance activities. [5 x 1]

Activity 1: _____ Activity 2: _____
 Activity 3: _____ Activity 4: _____
 Pick one kind of _____ maintenance activity and show an example of it in maintaining the *OmniEditor* Web Service _____

SOFTWARE REUSE

18. Why are Web Services good for software reuse? Please explain according to the five characteristics of software reuse: [5 x 1]

Classification: _____
Abstraction: _____
Selection: _____
Specialization: _____
Integration: _____

19. What is the difference between “build-for-reuse” and “build-with-reuse”? [1 x 2]

Build for reuse is to _____
Build with reuse is to _____

20. Which kind of following changes will you consider in order to reuse the *OmniEditor* web service for an *OmniGraphEditor* client? [2 x 2]

1. Add new operations to the *OmniEditor* WSDL, such as “uploadGraph” and “downloadGraph”
2. Generalizing message types of *OmniEditor*, such as from “upload(String)” to “upload(Object)” and from “download(String)” to “download(Object)”
3. Create a completely new web service *OmniGraphEditor*, with “upload(Graph)” and “download(Graph)” operations

Choice no. _____ Please explain the reason for the choice _____

21. In developing an editor client to reuse with an *OmniEditor* web service, how will you know whether it supports a centralized architecture (a client talks to the server) or a peer-to-peer (P2P) architecture (a client talks to its neighboring clients)? [2 x 2]

1. Looking at the WSDL.
2. Looking at the documentation of the Web Service.
3. Testing the Web Service with a centralized architecture use scenario and a peer-to-peer architecture use scenario.

Please pick one of the choices and indicate how does a WS client developer decide the architecture style of the web service? Choice no. _____
How? _____

Total Pages = (7), Total Marks = (100)