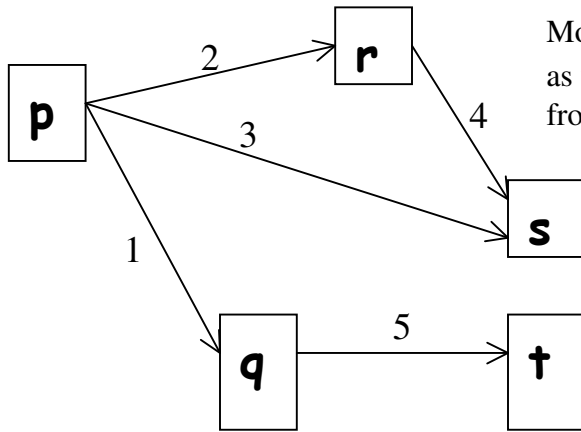


**1. Coupling Example (20 pts)**

Given a set a modules that print or store student status reports, with data-flow connections as shown in the diagram and table below (arrow direction is ‘In’ flow, and ‘Out’ flow is counter-arrow), determine the coupling type between modules (filling in the COUPLING TYPE table)

MODULE CALL/RETURN DATA		
number	In	Out
1	Student (complex object)	StudentValid flag
2	output function selector code	-
3	student id number	-
4	-	StudentReport (complex object)
5	student id number	student credit load



The file “**F . TMP**” is used by modules **p**, **s**, and **t**.  
 Modules **p** and **t** both write Student data to the file as well as read from it, while module **s** only reads from the file.

COUPLING TYPE (data, stamp, control, common, content)					
	<b>p</b>	<b>q</b>	<b>r</b>	<b>s</b>	<b>t</b>
<b>p</b>		(a)	(b)	(c)	(d)
<b>q</b>			(e)	(f)	(g)
<b>r</b>				(h)	(i)
<b>s</b>					(j)
<b>t</b>					

**2. True/False (10 pts):**

	Statement	T/F
a)	Java interfaces are useful for providing partial implementations of methods	
b)	White-box $C_2$ path test coverage ensures no bugs remain in a program	
c)	The key to meeting performance goals is early code optimization on a generic platform	
d)	Distributed application performance is independent of network effects	
e)	SCM systems resolve merge conflicts according to the Application Specification priorities	
f)	Informational cohesion is too weak for use with ADT implementation modules	
g)	A System/Application Acceptance Test Plan should be independent of the individual performance of each implementation module	
h)	According to the Stable Abstraction Principle, stable packages should be concrete	
i)	High Reliability implies High Availability	
j)	Pipe and Filter architecture provides good support for concurrent execution	

**3. Equivalence Classes (5 pts)**

You are given a method signature and need to generate test cases for it; the method signature is:  
`public float squareRoot(int x) throws IllegalArgumentException;`

List the input equivalence classes for testing this function:

List boundary values tests (input->expected response):

**4. Performance Analysis (10 pts)**

An application operates by receiving requests, creating a 5-MB size temp file for each request to hold interim results, then sending back the results and deleting the temp file. The processing time averages 12 seconds per request. Supposing you want to handle a throughput of 750 requests/sec, how much disk space should you expect the temp files to take, on average? (Your answer should be a simple calculation resulting in a size.)

**5. Robustness Analysis (5 pts)**

Given this sentence in a use case:

"The User adds his/her ID to the ClassListEntity."

What, if any, changes would robustness analysis require? (Your answer should a short sentence or two, and a rewritten bit of use case text..)

WORTH 8% of the class grade, so 80 points

ESTIMATED TIME: 3 hours

## 6. Lecture Review UML (20 pts)

“You may review a lecture for extra credit! Your review is graded 0 .. 3, just as a regular paper review and with the same criteria, but the points awarded are 1/2 the points of a required review. So you can get 0, 5, 8, or 10 points for writing up a lecture. Each student can only review each lecture once; each lecture can be reviewed by any number of students. (Would diagramming that make a good UML question for the Final Exam?)”

**Yes, it would ☺** ... but we'll only do part of it for a Midterm.

In the approach taken below, the TA is the central “Factory” ... the TA learns of Reviewable things (lecture notes, perhaps) from the external “Instructor” actor, and tells the Students about them; if/when a Student decides to submit a Review, they ask the TA for a blank Review for that Reviewable, fill out the Review, and submit it to the TA for scoring and record-keeping. The Student keeps track of the scores he/she has received. A Reviewable object tracks which Students have submitted Reviews of that Reviewable.

You are to draw out:

- (A) [10 pts] a UML static structure diagram including the classes shown below plus other classes you decide you need to capture the structural relationships described above (my guess is you'll have between 7 and 10 total classes). You may indicate cardinality constraints with multiplicity annotations on associations, and/or using Notes anchored to associations and/or constraints; or, of course, you may elect other UML mechanisms as well.
- (B) Two UML sequences covering the scenarios of:
- (1) [5 pts] a normal review sequence beginning with the Instructor creating a Reviewable opportunity (named ‘r\_able’) and notifying the TA of r\_able’s existence, followed by the TA telling a student named ‘s’ about r\_able, then s doing what he/she needs in order to submit a Review for grading. (The TA assigns it a ‘3’ and that’s the end of that sequence.)
  - (2) [5 pts] a later (asynchronous) attempt by Student ‘s’ to submit another Review for r\_able, which terminates when the TA assigns a score of ‘0’ to the rejected Review.

Note that in the sequence diagramming part you aren’t asked to show the complete lifecycle of a Reviewable and a Review and a Student, etc ... just get the Reviews generated and turned in. You do not, for example, have to deal with the eventual grading and reporting; nor do you have to sort out how the TA knows if the Reviewable is ‘assigned’ or is ‘extra credit’ ... the Instructor and the TA have some channel for that, all you need to do is show that the TA checks for that status at the correct time.

You may combine parts (B)(1) and (B)(2) on one sequence diagram (put a horizontal line on the diagram to separate the sequences) or you may do them as separate diagrams. You may need to create new methods on the implementation classes.

Modeling hint: Showing an Exception return can be done using an explicit ‘return value’ arrow labelled with the type (“:IllegalArgumentException”), or just an asynchronous message from the recipient back to the caller, as though the original message had returned but then the exception was thrown.

You are given the following class definitions to get started with:

```
interface Reviewable {
    // addReview throws an Exception to indicate the Student argument
    // is unacceptable; either an invalid or an already-seen Student
    void addReview(Student s) throws IllegalArgumentException;
    String getSubject();
}

interface Review {
    Reviewable getSubject();
    Student getAuthor();
    void setText(String); // add review content
    void setGrade0();
    void setGrade1();
    void setGrade2();
    void setGrade3();
    int getPoints();
}

interface Student{
    void doReview(Reviewable r); // creates Review and submits to TA for score
    int  getScore(Reviewable r); // score on single Reviewable
    int  getTotalScore(); // sum of all scores
}

interface TA{ // combines TA and ReviewFactory for simplicity
    void newReviewableExists(Reviewable r); //
    Review getBlankReview(Reviewable r); // Review factory behavior
    int receiveReview(Review r, Student s); // returns score
}
```

## 7. Parallelism (10 pts)

Suppose we have a program that runs on a single-processor machine in 10 seconds. The program has some parts which can be efficiently split into parallel processes across the network, but there is a part at the beginning and at the end which happens on just one processor (splitting data up for the parallel work, and combining the results.) On the single-processor machine, these two parts take 1 and 3 seconds, respectively.

You've been given a boatload of money to make this overall program run faster – you can add a lot of very fast parallel computers, and even use the new InfiniteSpeed/ZeroLatency network ☺ The single-CPU part of the program, though, has to stay on the existing computer (for security reasons). What's the shortest period of time you can hope to make this program take? Your answer should be a very short calculation invoking a well-known Law of operational research.

## Free Questions! (0 pts)

(0 pt) How do you get a development team to optimize for a particular characteristic?

(0 pt) What question took you the longest to answer, and how long was it?

(0 pt) How long did the entire test take you?

WORTH 8% of the class grade, so 80 points

ESTIMATED TIME: 3 hours

CMPS115 Winter 2004 Exam #2 OPEN BOOK/NOTE WORK ALONE (NO TEAMS)  
COVERS: Lectures 6 – 17

WORTH 8% of the class grade, so 80 points

ESTIMATED TIME: 3 hours