

CS3733-B03 Software Engineering

Final Exam

Name:

Answer all of the questions as completely as possible. Do your own work. There is an academic honesty policy at WPI and it will be enforced. Good luck.

Part 1: Basic Knowledge (60 pts.)

1. In the battleship game (or your Othello game), it is necessary to handle the event when a player clicks on a square. In battleship, I used a subclass of JButton for the squares and attached an event handler to handle the mouse click. The technique of handling an event by adding a listener to the button is an example of what design pattern?

Observer

2. What are at least three differences between the analysis model and the design model of a software product?

Design model is more detailed

Design model contains actual attributes and methods, the analysis model contains an overview of messages

Analysis model uses the entity, bounday, interface stereotypes

Analysis model deals with the problem space rather than the solution space

The analysis model basically breaks down use cases into an object model, the design model synthesizes the object model into a class and component hierarchy.

...

3. Provide three reasons why Clements and Parnas say that we will never see a software project that proceeds in the "rational" way.

(1) In most cases the people who commission the building of a software system do not know exactly what they want and are unable to tell us all that they know.

(2) Even if we knew the requirements, there are many other facts that we need to know to design the software. Many of the details only become known to us as we progress in the implementation. Some of the things that we learn invalidate our design and we must backtrack. Because we try to minimize lost work, the resulting design may be one that would not result from a rational design process.

(3) Even if we knew all of the relevant facts before we started, experience shows that human beings are unable to comprehend fully the plethora of details that must be taken into account in order to design and build a correct system. The process of designing the software is one in which we attempt to separate concerns so that we are working with a manageable amount of information. However, until we have separated the concerns, we are bound to make errors.

(4) Even if we could master all of the detail needed, all but the most trivial projects are subject to change for external reasons. Some of those changes may invalidate previous design decisions. The resulting design is not one that would have been produced by a rational design process.

(5) Human errors can only be avoided if one can avoid the use of humans. Even after the concerns are separated, errors will be made.

(6) We are often burdened by preconceived design ideas, ideas that we invented, acquired on related projects, or heard about in a class. Sometimes we undertake a project in order to try out or use a favourite idea. Such ideas may not be derived from our requirements by a rational process.

(7) Often we are encouraged, for economic reasons, to use software that was developed for some other project. In other situations, we may be encouraged to share our software with another ongoing project. The resulting software may not be the ideal software for either project, i.e., not the software that we would develop based on its requirements alone, but it is good enough and will save effort.

CS3733-B03 Software Engineering Final Exam

4. In a couple of sentences, tell what Clements and Parnas advise you to do in order to overcome the problems of not proceeding in a rational way?
They suggest that you take time at the end of the project to make it look like you followed the process. You create the appropriate documents and other artifacts that you would have done if you had followed the process the way you would have if everything were perfect.
5. What is a design pattern?
A design pattern is a common solution to frequently encountered problems.
6. There are three types (levels) of testing. What are they? Which one primarily would use use cases to drive the test cases?
Unit testing, integration testing, system testing. System testing is the most use-case driven.
7. What is exploratory testing? What are the two primary results of exploratory testing?
Exploratory testing is a systematic way of exploring a software product. During exploratory testing you will identify defects and come up with ideas for further testing of the product.
8. What type of UML diagram is used to describe a specific scenario of a use case?
Sequence diagram.
9. What are the two interfaces for EJBs and what is the purpose of each?
The Home interface is used for creation, deletion, etc. (lifecycle methods).
The Remote interface is used for the actual logic and business rule implementation.
10. What is the environment (context) in which EJBs execute, and what are at least two things that the environment provides?
The environment is called the Container. It manages transactions, persistence, security, distribution, etc.

Part 2: Application (40 points)

1. Assume you were going to review the following code. You are to inspect it for consistency in naming, formatting, and understandability of style. You also are going to review it for correctness. Identify all of the things you find that you would raise during a code review. Identify each item by specifying the line number (if appropriate) and the problem.

CS3733-B03 Software Engineering Final Exam

```
1  /**
2   * Method to return the type of triangle represented by three sides.
3   * This only checks for whether a triangle is equilateral, isosceles, or
4   * scalene.
5   * @return a string with a description of the type of triangle
6   * @param side_a the first side
7   * @param side_b the second side, etc.
8   */
9  public String triangleType(int side_a, int b, int c)
10 {
11     if (side_a == b) return TRIANGLE_ISOSCELES;
12     if (side_a == c) {
13         return TRIANGLE_ISOCELES;
14     }
15     if (side_a == b && b == c)
16     {
17         return EQUILATERAL_TRIANGLE;
18     }
19     return SCALENE_TRIANGLE;
20 }
```

(2-pts. for each of the first six, and 4 pts. each for the last two)

- Only two parameters are identified in the javadoc comments.
 - The comment says side_b, but the argument is b.
 - The arguments should be named consistently, either side_a, side_b, side_c, or a, b, c.
 - Using the underscore in side_a and the mixed-case method name, triangleType, are inconsistent naming.
 - Inconsistent use of braces on the if statements.
 - Inconsistent naming of return types, use either TRIANGLE_... or ..._TRIANGLE.
 - The method will never return EQUILATERAL_TRIANGLE.
 - Never checks for side b and c equality.
2. You are designing a class that handles login and security for an application. The basic functions of the class are:
- a. The user must first enter a valid login name. The system checks to see that the name is valid before asking for a password. If it is not valid, the user can either exit (cancel) or re-enter the name.
 - b. The user then must enter the password associated with the name. The system validates that the password is the appropriate one for the login name. If the password is invalid, the user can either cancel and exit, re-enter the password, or go back to step (a) and re-enter the login name.
 - c. The user selects a function. The system validates that the function is one that the user is allowed to do. If not, the user can either exit, re-enter the function, or go back to step (a) and start the process over again.

Create a state diagram that describes the above actions. You should name each state (except the start and final state) and the events that trigger the transitions from state to state.