

ISDS 406 --- Final Exam Study Guide

Logistics: Unless you have arranged a special agreement with the instructor, you must take the exam on the date and time assigned to your class section.

The exam will be administered via TITANium, but you must take the exam using one of the classroom computers. You may not use any notes or computer programs. You may not sit next to someone on your team.

Point value: The exam is worth 175 points.

Breakdown: 85% multiple choice + 15% Free Response

Multiple Choice portion (37 questions):

- 4 questions from Chapter 7 (Design Phase / Acquisition Strategies). You will need to know:
 - What are the steps of the design phase, and what is the final deliverable of the design phase?
 - What are the three acquisition strategies? Under what circumstances would you choose each one?
 - What is an RFP? RFI? RFQ? What are their purposes?
 - Alternative matrix: what is listed across the top row and what is listed down the first column? How is the right alternative selected in a matrix? (hint: the one with the highest weighted score)
- 6 questions from Chapter 8 (Architecture Design). You will need to know:
 - Definition of architecture design
 - Two major architectural components (hint: see next bullet point)
 - What are the four basic functions of software? What are the three primary hardware components?
 - In a client-server architecture, which of the four software types usually resides on the client, which usually resides on a server, and which varies depending on the architecture chosen?
 - In a two-tiered client-server architecture, what is the difference between a thick and thin client?
 - In mobile architecture (where the client is a mobile device), what is a rich client and what is a thin Web-based client? What technologies are used to create cross-platform apps (hint: Web-based)
 - What is cloud computing, and can you name an example of cloud computing?
 - HW-SW specification: purpose and format (what goes in the columns and what are the typical rows)
 - You will not need to know: virtualization, sub-categories of nonfunctional requirements
- 7 questions from Chapter 9 (User Interface Design). You will need to know:
 - What are the three key mechanisms of the user interface?
 - What is the one key driving concept of a good user interface design?
 - Definitions of the 6 principles of user interface design
 - Three layout areas (see "Layout" slide from lecture)
 - For which type of user should you focus on ease of use? For which type of user should you focus on ease of learning (hint: see "Usage level" slide from lecture)
 - What is the 3 clicks rule and which of the 6 user interface design principles does it correspond to?
 - What are the 5 steps in UI design process?
 - What is an interface structure diagram (ISD)?
 - What is an interface metaphor? Be able to recognize one
 - What are the types of prototypes?
 - What is source data automation? Know examples

- You don't need to know: touch screen design, personas, interface evaluation methods, menu tips, message tips and types of messages, input validation types, lookups/dropdowns/default values, output design principles
- 4 questions from Chapter 10 (Program Design). You will need to know:
 - Definitions and purposes of structure charts, program specifications, and pseudocode
 - Difference between logical and physical process models
 - Steps to create a physical DFD
 - You don't need to know: in-depth details of structure charts and program specifications; pages from Ch 10 that were not assigned reading; Ch 11 (physical *data* models)
- 6 questions from Chapter 12 (Moving into Implementation). You will need to know:
 - Definition of "Implementation" – what happens during the Implementation Stage?
 - Best practices for managing programmers
 - What are the three work areas for programming?
 - What are the 4 categories of testing? Define each
 - Black box vs white box testing
 - In which of the 4 categories do you test non-functional requirements?
 - Alpha vs beta testing
 - What are the two types of documentation?
 - What are the five types of documentation navigation (hint: one of them is table of contents)
 - You don't need to know: slide about "people involved in implementation", change control and program logs, classic mistakes, five common functions to test in unit testing, types of integration testing, specific types of "system" testing, three types of user documentation, three ways to organize documentation, writing guidelines for crafting documentation topics
- 6 questions from Chapter 13 (Transition to New System). You will need to know:
 - Conversion strategies:
 - Direct vs. parallel
 - Pilot vs. phased vs. simultaneous
 - Whole system vs. module-by-module
 - 3 key factors that determine which combination of conversion strategy you select
 - Which strategy is best for high risk / low risk / high cost / low cost / limited time / plenty of time
 - What is a business contingency plan?
 - What are the 4 steps of change management?
 - Categories of potential adopters
 - What is the correct focus of training on a new system? (hint: it is NOT trying to explain every system feature)
 - What are the 3 important POST-implementation activities?
 - What are Level 1 and Level 2 support?
 - You don't need to know: "elements of a migration plan", management tools for supporting adoption, sources of maintenance change requests, details about project assessment
- 4 questions from Chapter 14 (Object-Oriented Approach). You will need to know:
 - Definitions of "object-oriented approach", class, object, instance, UML
 - What is the methodology created specifically for the OO approach?
 - What are the two major groupings of UML diagrams?
 - What are the four fundamental UML diagrams, and which of the four serves as the basis or starting point for the rest?
 - You will not need to know: complex use case diagrams (e.g., "includes" and "extends"), how to read any UML diagrams beyond use case diagrams

Free Response portion (4 questions): For each of these questions, I will show one example document or diagram. You will be required to (a) name what type of document or diagram it is, and (b) state at least 3 things that you see wrong with the document (e.g., “button sizes in the prototype are not consistent”, etc.). Therefore, you must understand the key elements and guidelines for the following types of documents and diagrams that analysts and designers create (or are involved with):

- Acquisition Strategy Recommendation
- Hardware-Software Specification
- Interface Structure Diagram
- User Interface Prototype (use the 6 design guidelines to critique)
- Physical Data Flow Diagram
- Use Case Diagram