CS2212

Introduction to Software Engineering

Final Exam Overview

Ask Questions Live

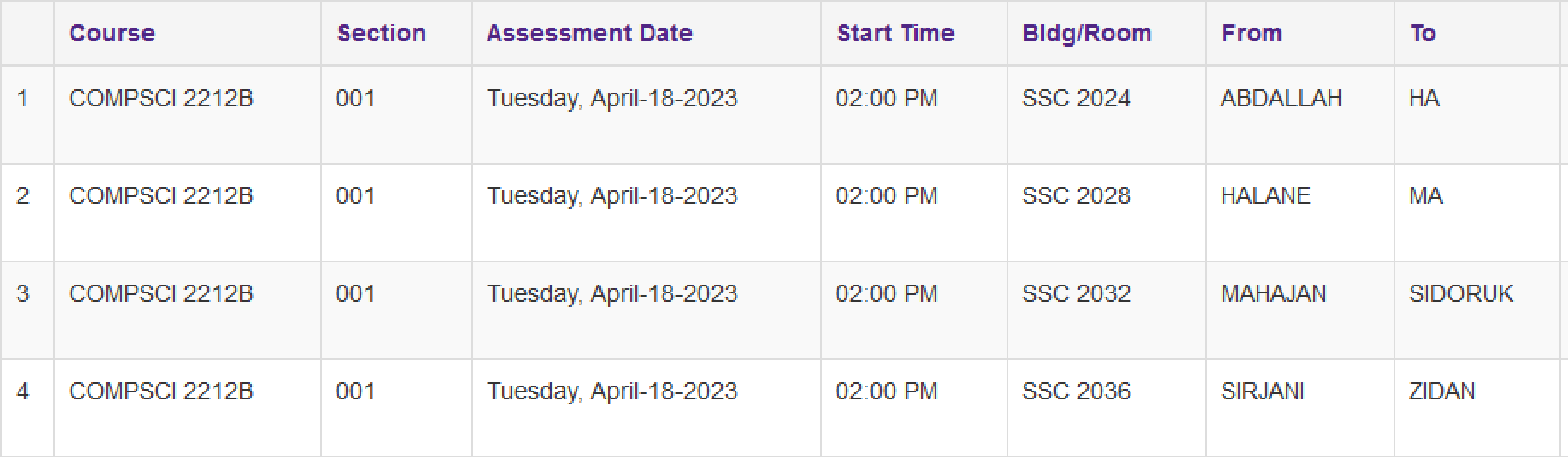
cs1.ca/ask

Some images and content from Software Engineering: A Practitioner's Approach 9th Edition By Roger Pressman and Bruce Maxim

Final Exam

Location/Time

- **Date:** Tuesday, April 18th at 2PM
- **Location:** SSC (Social Science Centre)
- **Length:** 3-hours
- **Room Based on Last Name:**



	Course	Section	Assessment Date	Start Time	Bldg/Room	From	To
1	COMPSCI 2212B	001	Tuesday, April-18-2023	02:00 PM	SSC 2024	ABDALLAH	HA
2	COMPSCI 2212B	001	Tuesday, April-18-2023	02:00 PM	SSC 2028	HALANE	MA
3	COMPSCI 2212B	001	Tuesday, April-18-2023	02:00 PM	SSC 2032	MAHAJAN	SIDORUK
4	COMPSCI 2212B	001	Tuesday, April-18-2023	02:00 PM	SSC 2036	SIRJANI	ZIDAN

Final Exam

Mixed Format:

- **Multiple Choice** (75 questions, 75 points total, 1 point each)
  - Primarily textbook content.
- **Short Answer** (15 questions, 75 points total)
  - Points per question will depend on question type.
  - Some questions may have multiple parts.
  - Primarily software engineering work products (diagrams, use cases, etc.), calculations, and simple JavaDoc and JUint code.
- Total 90 questions, 150 points.

Final Exam

Content

- **Comprehensive:** everything from textbook, slides, videos, tutorials, and in-class activities.
- **Textbook Chapters:**
  - **Chapter 1:** Software and Software Engineering
  - **Chapter 2:** Process Models
  - **Chapter 3:** Agility and Process
  - **Chapter 4:** Recommended Process Models
  - **Chapter 7:** Understanding Requirements
  - **Chapter 8:** Requirements Modeling
  - **Chapter 9:** Design Concepts
  - **Chapter 10:** Architectural Design
  - **Chapter 11:** Component-Level Design

Chapter 12: User Experience Design

Chapter 14: Pattern-Based Design

Chapter 15: Quality Concepts

Chapter 16: Reviews

Chapter 17: Software Quality Assurance

Chapter 19: Software Testing (Component)

Chapter 20: Software Testing (Integration)

Appendix 1: An Introduction to UML

Final Exam

Multiple Choice Questions:

- **Some important topics:**
- Software engineering definitions
- Software application domains
- Software engineering layers
- Generic process framework
- Framework and umbrella activities
- Process flows
- Stakeholders
- Prescriptive process models
- Agile process models
- Recommended process model
- Actors (primary/secondary, active/passive)
- Use cases & User stories
- Functional & non-functional requirement
- Requirements engineering tasks
- Conflict resolution technique
- Grammatical parses
- UML diagrams (e.g. what one to use for a specific case).
- Requirements models
- Design models
- Abstraction
- Coupling & cohesion
- Refactoring
- Separation of concerns
- Information Hiding
- Software components
- Architectural styles
- Patterns & anti-patterns
- Architectural Context Diagram
- Archetypes
- Basic design principles (e.g. Interface Segregation Principle, Dependency Inversion Principle, etc.)
- Mandel's three golden rules for User Interface Design
- UX and UI
- UI metaphors
- Customer journey maps and user personas
- Task analysis
- Google's 5-Day UX design sprint
- Useability guidelines
- Views of quality
- ISO 25010 standard
- Case studies of quality (e.g. Animusic)
- Quality costs
- Review metrics
- Reviews and postmortem evaluations
- Errors and defects
- Loop testing
- Basis path testing
- Unit testing
- Integration testing
- Validation testing
- System testing
- Equivalence partitioning
- Cyclomatic complexity
- White/Black box
- Verification & validation
- Smoke testing
- Regression testing

Final Exam

Questions will involve: drawing diagrams, filling in templates, writing usage scenarios (use case, user stories), writing JUnit and JavaDoc code, and calculations (e.g. cyclomatic complexity).

Short Answer Questions:

- **Some important topics:**
  - UML activity diagrams (used for both representing algorithms and use cases).
  - Flow graphs and Cyclomatic Complexity
  - Basis path sets
  - JavaDoc comment (should know common tags: @return, @param, @throws, @see, @author, etc.)
  - JUnit (should know how common asserts work: assertEquals, assertTrue, assertThrows, assertArrayEquals, etc.)
  - UML class diagrams (including relationships between classes, data types, method parameters, public/private, multiplicity, etc.).
  - UML use case diagrams (including generalizations between actors).
  - Use cases (template will be given)
  - Actor descriptions (template will be given)
  - User stories and acceptance criteria
  - Anything from your project’s requirements, design, and testing documents

Final Exam

Rules

- No electronics (calculators, phones, smart watches, etc.).
  - Phones must not be on your person (in backpack or left at home/car). Having a phone on your person will be considered cheating.
  - Smart watches not allowed (must be in backpack or left at home/car).
  - No calculators (should not need one).
- Must arrive on time (cannot write if more than 30 min late)
- Must have student card with you.
- Closed book but one page cheat sheet allowed (handwritten, letter size, two sided).
- Normal exam rules (no talking, cheating, etc.)
- Cannot leave in the last 30 minutes (have to wait till end of exam).
- Do not alter the barcode on the short answer booklet or write in the top inch of the page.
- All booklets, cheat sheets, scrap paper, etc. must be returned at end of exam.

Final Exam

Time Management

- **Multiple Choice:** about 1 minute per question, aim for less than ½ the exam time in total.
- **Short Answer:** will vary for each question, aim at least ½ the exam time in total.

YOU'VE GOT THIS!

Western
StudentExperience

Check out these academic supports from Student Experience to help you get through finals season.

LEARNING DEVELOPMENT & SUCCESS

- Get personalized academic support
- Learn study skills & exam strategies
- Practice tips for stress management

learning.uwo.ca

WRITING SUPPORT CENTRE

- Get personalized support
- Learn to write persuasively
- Strengthen your final assignments

writing.uwo.ca