

# Exam Prep Document

## Exam Format

- Sections:
  - Very short answer
  - Short answer
  - Applying concepts: design and analysis
- Types of questions: Similar questions to reviews at beginning of lecture, e.g., vocabulary, comparisons, fix the code, explain the code, ...; synthesizing concepts/content
- On Canvas
  - Part will be “in Canvas”
  - Part will be in Word – download the Word file, complete/fill out
- Two hours to complete the exam on Friday between 9 a.m. to 11:59 p.m.
- It is open notes, slides, book, and mind. Closed everything else. If you are looking up every question (or even many questions), you will not complete the exam.

## What I expect from you

- Be precise, clear in your answers.
- Do not add explanations for things I did not ask for
- Make sure I know that you know what you’re talking about.
- To be comfortable with the terminology/vocabulary of web applications and software engineering to understand it when reading it and to use the appropriate terminology when describing/discussing it.
- Know the what, why, and how of concepts (so I won’t include this in each of the topics below)
- Understand tradeoffs
- You can use bulleted lists or tables, as appropriate for the question.
- To design a solution and be able to describe/explain its design
- Syntax must be very close to correct (correct keywords, punctuation, special characters, variable naming, operations)
- To be able to recall the information without looking up every (or even more than a few) question. If you need to look up answers, you will not complete the exam in time.
- To keep an eye on the time remaining.
- To consider the point value of the question. If the question is worth 4 points and is an “essay” question, I’m only looking for a few clear sentences/bullet points.

## What I do not expect from you

- Do not need to know the Image sharing article or the Google search article

## How to prepare

- Review slides (including review questions), readings
  - Practice using the terminology – that’s why we discuss and you articulate the ideas!
- Review assignments
- Practice reading and writing HTML, CSS, Servlets, JSPs, SQL queries, JDBC, JavaScript

# Topics

## WWW Architecture

- Components – Internet, clients, servers
- HTTP protocol
- Process of requesting a Web page

## HTML5

- Terminology (elements, tags, attributes, content), components, syntax, purpose, rules

## CSS

- Terminology (selector, properties, values), components, syntax, purpose, rules
- Bootstrap

## Web application servers

- Handle requests (get, post)
- Return responses
- Servlet container
- Handling state
- Multi-threaded → synchronization
- How different from web server

## Java Servlets

- Important methods: init, doGet, doPost, destroy
- Sessions
- Session vs Request vs Context (Application) attributes
- Parameters vs Attributes vs Init Parameters
- Maintaining state across multiple requests: hidden fields, cookies, session state
- Organization

## JSPs

- Syntax
- Organization with servlets

## JavaScript

- Syntax, rules
- Understanding code

## Databases, SQL, JDBC

- Why DB? Why SQL? Why JDBC?
- Terminology – attributes/columns, rows, queries, joins
- Organization of tables, attributes, data types
- Common SQL statements, meanings

## JDBC

- Why JDBC?
- How to connect
- Basic Syntax

## Software Engineering Tools

- Version control: purpose, benefits, limitations
- Maven: purpose, benefits, limitations

## Quality Attributes

- How is web software different from other software?
- How do those characteristics change what constitutes “quality software” on the Web?

## Usability

- Fundamental ideas
- Guiding principles
- Trunk test
- Accessibility

## Security

- Why the web is a target
- Design principles
- SQL injection attacks
- Cross-site scripting

## Other Suggestions

- Focus on the most important/distinguishing characteristics/traits
  - Example: if there is a 4-point question is “What is JSP? Explain a benefit of JSPs.”, do not start copying all of the content from any slides you can find about JSPs. You should know the highlights and be able to summarize what a JSP is, e.g., “A JSP is a Java Server Page. JSPs are like an HTML page with dynamic components, generated by Java code.” Followed by “Benefit: JSPs are mostly written in HTML with small snippets in Java, which, as opposed to servlets, makes it easier to focus on the *view/presentation* of web pages in HTML.”
  - Example: if the question is “What are the benefits of using JSPs?”, do not answer with any variation of “Because JSPs are beneficial” or “Because JSPs generate web pages.” Tell me *why* JSPs are good. You want to make clear that you don’t just know the terms but you also understand their meaning and implications.