**Title: Learning goals**

**Topic: Overview**

Anticipate the structure of CPSC 101 lectures\ and exams based on their learning goals

**Title: What is a computer**

**Topic: Overview**

Categorize a well-described device as a computer or non-computer with clear, articulate reference to (for one pass) your own explicit definition and (for another) our course?s agreed definition\

**Title: Define algorithm**

**Topic: Overview**

Define the term algorithm

**Title: Algorithms in new domains**

**Topic: Overview**

Relate the components of an algorithm to a new domain

**Title: Define data representation**

**Topic: Data representation**

Define a data representation scheme

**Title: Properties of data reps**

**Topic: Data representation**

describe properties of data representation\ schemes that can be found in many contexts of\ the world around you

**Title: Critique of data reps**

**Topic: Data representation**

critique properties of data representation\ schemes, from the stand-point of usability and\ engineering considerations, given information\ about the context in which the scheme is used

**Title: Design of data reps**

**Topic: Data representation**

engage in design of data representation schemes, for example by proposing modifications that address shortcomings of given data representation schemes

**Title: Practical data reps**

**Topic: Data representation**

put your knowledge of data representations to practical use, for example, in making decisions about representing your own data, or deciding you want to go into CS, or applying the knowledge in your own field

**Title: Tools augment thought**

**Topic: HCI**

explain how tools augment and constrain our power to think and act

**Title: Myth of human error**

**Topic: HCI**

define the ?myth of human error?

**Title: Critique interfaces**

**Topic: HCI**

use the concepts of negative transfer, familiarity/consistency, good mapping and metaphors, and useful feedback to explain strengths and weaknesses of simple interfaces

**Title: GUI vs. CLI**

**Topic: HCI**

compare the strengths and weaknesses of graphical vs. command-line user interfaces

**Title: Layered design**

**Topic: Networks**

explain why breaking a system into layers can make it easier to understand and build applications at each layer

**Title: Terms**

**Topic: Networks**

define ?IP address?, ?domain name?, and ?URL?

**Title: Term Rel'nships**

**Topic: Networks**

explain the parts of and relationships among IP addresses, e-mail addresses, domain names, and URLs

**Title: Communication over the Internet**

**Topic: Networks**

trace the process of communication from one computer to another through the layers of the internet

**Title: Network types**

**Topic: Networks**

define ?point-to-point? and ?broadcast? networks

**Title: Point-to-point routing**

**Topic: Networks**

justify key elements of a point-to-point routing protocol

**Title: Broadcast protocol**

**Topic: Networks**

justify each step of the party protocol for broadcast networks

**Title: Search engine parts**

**Topic: Web Pages and Search Engines**

briefly describe the parts of a web search engine

**Title: Indexing**

**Topic: Web Pages and Search Engines**

explain how a search engine finds and indexes web pages

**Title: Crawling a page network**

**Topic: Web Pages and Search Engines**

predict how and whether a page will be found by a web crawler given the link structure around the page and which pages the crawler already knows of

**Title: HTML coding**

**Topic: Web Pages and Search Engines**

construct HTML to present the structure and visual appearance you intend for an HTML webpage

**Title: How to learn programming concepts**

**Topic: JavaScript**

apply the ?study/model/predict/experiment/refine? technique to learn new programming concepts

(Note: study a topic, model how it works in your mind, predict its behaviour on a set of common and unusual problems, experiment with what actually happens (which we can often do easily in computer science!), refine your model, and repeat.)

**Title: Terms and use**

**Topic: JavaScript**

connect the following terms to their use in programs: function, function declaration, function call, function name, function body, parameter, parameter list, variable, value, statement, return statement, assignment statement, expression, orthogonality, sequential execution. (This is a BIG goal mostly accomplished through textbook reading and practice problems.)

**Title: Programming concept: variables**

**Topic: JavaScript**

accurately model and predict the behaviour of variables in JavaScript programs

**Title: Programming concept: sequential executio**

**Topic: JavaScript**

accurately model and predict the flow of control in a JavaScript program through sequential execution

**Title: Orthogonality**

**Topic: JavaScript**

employ the concept of 'orthogonality' to combine your knowledge of different programming elements, including being able to interpret their combined effect

**Title: Programming concept: expressions**

**Topic: JavaScript**

accurately model (and so predict) the evaluation of any expression, no matter how complex, as long as you have a good model of the parts

**Title: Programming concept: expressions, part 2**

**Topic: JavaScript**

model the construction and interpretation of expressions using numbers, strings (text snippets in quotes), arithmetic operators, and function calls

**Title: HTML <script> tag**

**Topic: JavaScript**