Review for Quiz

November 14, 2016

Info

http://docs.cs50.net/2016/fall/quiz/about.html

- 48 hour window in which to take the quiz.
  - You should require much less than that; expect an appropriately-scaled down version of the Test.

- Released Tue 11/15 at noon, due via submit50 Thu 11/17 at noon.
  - Be sure to run `update50` in your IDE before submitting!
  - Submitting seven minutes late is equivalent to not submitting at all; don’t wait until the last possible second.

Info

http://docs.cs50.net/2016/fall/quiz/about.html

- Any topic that we have covered in the entire course is fair game, including lecture at Yale.

- The quiz will nonetheless be focused primarily on content from Weeks 6 through 10, inclusive, and Problem Sets 6 through 8, inclusive.

- Staff will be returning the quizzes next week (after Harvard-Yale).

Resources

- Review lecture notes.
- Review lecture source code.
- Review lecture slides.
- (Re)watch lecture videos.
- Review problem set specifications, distribution code, and postmortems.

Resources

- Office hours
  - Today (Monday) at HSA through 5:30pm.
  - Tonight in Widener/AKW; consult OH calendar.

Resources

- Office hours
  - Today (Monday) at HSA through 5:30pm.
  - Tonight in Widener/AKW; consult OH calendar.

- No office hours during the Quiz (11/15 through 11/17) or indeed for the remainder of the semester!

Resources

- CS50 Discuss
  - You may post questions through the end of the day today.
  - You may not post questions on Discuss from Tue 11/15 through Thu 11/17.
    - Staff will not respond to any questions during this time, but will monitor the forum.

Resources

- CS50 Discuss
  - You may post questions through the end of the day today.
  - You may <u>not</u> post questions on Discuss from Tue 11/15 through Thu 11/17.
    - Staff will not respond to any questions during this time, but will monitor the forum.
- The only humans to which you may turn for help during the Quiz are the course's heads.
- The course heads will answer only logistical questions that arise during the Quiz, and will not answer content-related questions.

Resources

- CS50 Quiz Bank
  - Available at quizbank.cs50.net.
  - No login required.
  - Archive of old quiz questions from 2007—15, searching by topic and keyword.

Resources

- CS50 Quiz Bank
  - Available at quizbank.cs50.net.
  - No login required.
  - Archive of old quiz questions from 2007—15, searching by topic and keyword.

- Test, Fall 2016
  - Still available at cdn.cs50.net/2016/fall/test/test.html, along with an answer key (see David's email from after the Test was finished).

Weeks 0-5

- For all this material, consult the Test review session video from October 14, available at cs50.ly/testreview.

- The Quiz will have a heavier emphasis on content from Week 6 onward, but all of this material remains fair game.

Week 6

- Networking
  - “The internet” is a connection of a number of smaller *physical* networks, such as those in your home, to one another.
  - Services run on top of the internet, relying on the physical framework to move information along.

Week 6

- Networking
  - “The internet” is a connection of a number of smaller *physical* networks, such as those in your home, to one another.
  - Services run on top of the internet, relying on the physical framework to move information along.

- Routers
  - A physical device (typically inside your home or workplace) that directs information to the different devices connected to your local network.
  - Routers support different *protocols* (types of software) that help provide your devices with internet connectivity.

Week 6

- DHCP
  - *Dynamic Host Configuration Protocol*
  - Software that runs on your router that assigns your machine a (ostensibly) unique identifier—an IP address—so that other internet-connected devices can refer to it.

Week 6

- DHCP
  - *Dynamic Host Configuration Protocol*
  - Software that runs on your router that assigns your machine a (ostensibly) unique identifier—an IP address—so that other internet-connected devices can refer to it.
- IP addresses
  - Most commonly seen today as IPv4: *w.x.y.z* (each from 0-255).
  - Trending towards IPv6: *s:t:u:v:w:x:y:z* (each from 0000-ffff).
  - Public IPs and private IPs.

Week 6

- DNS
  - The “yellow pages” of the internet, translating the IP addresses belonging to servers for websites we might visit.

Week 6

- DNS
    - The “yellow pages” of the internet, translating the IP addresses belonging to servers for websites we might visit.

- `traceroute`

Week 6

- DNS
  - The “yellow pages” of the internet, translating the IP addresses belonging to servers for websites we might visit.
- `traceroute`
- TCP/IP
  - *Transmission Control Protocol* over *Internet Protocol*.
  - Break our internet requests down into small chunks (packets), route them from one machine to another (IP), then sort and assign those packets to the right service on the target machine (TCP).

Week 6

- Ports
    - Associating internet traffic with a “service” or “program” on a destination machine, which knows how to interpret that data.

- Common ports:
    - FTP – *file transfer protocol*: port 21.
    - SSH – *secure shell protocol:* port 22.
    - SMTP – *simple mail transfer protocol*: port 25.
    - DNS – *domain name service:* port 53.
    - HTTP – *hypertext transfer protocol*: port 80.
    - HTTPS – *secure hypertext transfer protocol*: port 443.

Week 6

- Firewalls
    - Software that runs on your machine that can block traffic to specific addresses (sites) or via certain ports.

Week 6

- Firewalls
  - Software that runs on your machine that can block traffic to specific addresses (sites) or via certain ports.

- VPN
  - *Virtual Private Networks*
  - Service that allows you to basically connect a “tunnel” between you and your VPN provider, and rely on your VPN provider to handle routing all of your traffic.
  - Can be used to get around firewalls.

Week 6

- GET
  - Simplest form of an HTTP *request*. Specify what page we want, the version of HTTP is using, and the host we are asking this of (plus perhaps additional info).
  - Should then receive an HTTP *response*, which specifies the HTTP version, whether the request succeeded, and then some HTML that corresponds to my request.

- Status Codes
  - 200, 301, 401, 403, 404, 500…

Week 6

- HTML
  - *Hypertext Markup Language*
  - The language of websites; text on our pages, arranged between *tags* that we open and close to provide some additional meaning for text… marking it up.
  - Not a programming language.

Week 6

- HTML
  - *Hypertext Markup Language*
  - The language of websites; text on our pages, arranged between *tags* that we open and close to provide some additional meaning for text… marking it up.
  - Not a programming language.
- CSS
  - *Cascading Style Sheets*
  - A language that defines the look and feel of our web pages, setting selectors and attributes to modify our web pages.
  - Not a programming language.

Week 7

- Machine Learning
  - A sub-discipline of computer science focused on making machines more helpful to humans by reducing the amount of time we need to interact with them (as by programming), but still allowing them to solve new problems.
  - Our programming is thus focused on making computers more intelligent so they are capable of honing their own skills, rather than on programming a new skill outright.

Week 7

- Image Classification
  - A large part of devising good machine learning algorithms is to think about the data sets that you will use to *train* your algorithm.
    - Size, quality, organization…
  - Can start out easy, categorizing black and white images of numbers as a grid of different contrasts.
    - Then we just use a multi-dimensional form of the distance formula to estimate how a test point matches against our training set.
  - Can get more complex, with color images, images being rotated, different perspectives.

Week 7

- Deep Learning
  - Pattern recognition on a grand scale.
  - Take pixels and abstract them into lines/edges.
  - Then take a collection of those edges and figure out how they fit into subcomponents of objects, then take those and combine them together to make complete objects.

Week 7

- Deep Learning
  - Pattern recognition on a grand scale.
  - Take pixels and abstract them into lines/edges.
  - Then take a collection of those edges and figure out how they fit into subcomponents of objects, then take those and combine them together to make complete objects.

- Siri, Cortana, Google Now: all of these services rely on deep learning—processing data, taking small pieces, combining them together, and figuring out based on common patterns what you might be asking for.

Week 7

- Text Clustering
  - We did a very simplified version of this in Problem Set 6, categorizing words/phrases/tweets into 3 groups.
  - Analyzing text by counting frequency (or normalized frequency) of appearance in various strings.
  - Depending on how many types of words we are seeking (how many dimensions), we can cluster similar text together similar to how we processed images.

Week 8

- Python
  - The tools and fundamentals remain the same, but the syntax has changed.
  - Variables, conditionals, loops, lists.
  - Tuples, sets.
  - Dictionaries.
  - Objects/classes.

Week 8

- Python
  - The tools and fundamentals remain the same, but the syntax has changed.
  - Variables, conditionals, loops, lists.
  - Tuples, sets.
  - Dictionaries.
  - Objects/classes.

- Interpreter
  - We needed to *compile* our programs in C in order to execute them, but in Python we can run an interpreter to consolidate that compilation step.

Week 8

- Higher-level languages
  - One of the clearest changes from C to Python and other modern languages is the ability to work at a higher level.
  - Things that were complex in a lower-level language (like a hash table) is a built-in piece of functionality in languages like Python.

Week 8

- Higher-level languages
  - One of the clearest changes from C to Python and other modern languages is the ability to work at a higher level.
  - Things that were complex in a lower-level language (like a hash table) is a built-in piece of functionality in languages like Python.

- Web-based software
  - Ultimate goals are to do things like:
    - dynamically generate HTML.
    - respond to and process data passed via HTTP GET or POST.
    - abstract out common, repetitive aspects.

Week 8

- MVC
  - *Model, View, Controller* paradigm for organizing the various aspects of your website.

  - Controller: All the logic of your program lives in these files; decision-making takes place here.
  - View: The look and aesthetics of your site; the pages (e.g., the HTML) that the user sees.
  - Model: The data on your site; usually hidden from direct modification by the user—requests for the data flow through the controller.

Week 8

- Flask
  - A web *microframework* for writing simple web applications.
  - The writers of Flask implemented some of the more tedious aspects of web development:
    - How to handle page redirection.
    - How to render—generate—page templates in HTML.
    - Code for determining the URL of a particular page.
  - Can use Python *decorators* to associate the behavior of a method with a route (URL) that the user of the site might visit.

Week 9

- SQL
    - *Structured Query Language*
    - A language that is used for making requests, also known as *queries*, to databases.
    - Unlike HTML and CSS, is a programming language.
    - SQL is <u>not</u> the database itself.
        - MySQL, PostgreSQL, SQLite, Access, Oracle
    - SQL databases are relational; they provide us with the ability to establish and maintain relationships across different *tables*.
    - CRUD – create, read, update, and delete data.

Week 9

- SQL Syntax
  - SELECT (*, FROM, WHERE)
  - INSERT (INTO, VALUES)
  - UPDATE (WHERE, SET)
  - DELETE (WHERE)
  - Primary keys, unique columns, NULL/not NULL, autoincrement.

Week 9

- SQL Syntax
  - SELECT (*, FROM, WHERE)
  - INSERT (INTO, VALUES)
  - UPDATE (WHERE, SET)
  - DELETE (WHERE)
  - Primary keys, unique columns, NULL/not NULL, autoincrement.

- phpLiteAdmin
  - We could manage our entire SQLite database infrastructure via the command line, but it can be really tedious sometimes.
  - phpLiteAdmin is a web-based tool for maintaining our SQLite databases using a graphical user interface.

Week 9

- Joining tables
  - As we put data into our tables, we might discover a lot of redundancy that bogs down our database.
  - Because of the relational nature of our SQL databases, we can “factor out” common information and share it across tables in the same database.
  - ZIP codes.

Week 9

- Joining tables
  - As we put data into our tables, we might discover a lot of redundancy that bogs down our database.
  - Because of the relational nature of our SQL databases, we can “factor out” common information and share it across tables in the same database.
  - ZIP codes.

- ORM
  - *Object-relational mapper* – example: SQLAlchemy.
  - Use a language like Python (instead of raw SQL) to build your database and its initial data, further abstracting the details out.

Week 9

- SQL Injection attacks
  - Remember that buffer overflow attacks exploit the nature of the stack in C to perhaps redirect user’s code.
  - SQL injection attacks exploit the nature of SQL syntax to try to trick the database into executing a query the programmer did not intend to permit.
  - Defend against this by *escaping* user input to prevent dangerous SQL queries.

Week 10

- JavaScript
  - The tools and fundamentals remain the same, but the syntax has changed… again!
  - Variables, conditionals, loops, arrays, objects.
  - Anonymous functions.

Week 10

- JavaScript
  - The tools and fundamentals remain the same, but the syntax has changed… again!
  - Variables, conditionals, loops, arrays, objects.
  - Anonymous functions.

- Events
  - “Event handlers” are listeners—functions—that are executed in response to *something* happening on our websites (mouse clicks, key presses, a form submission).
  - Such handlers tend to be anonymous functions.
  - Similar in concept are *callbacks*.

Week 10

- Writing JavaScript
  - JavaScript code can go inside of `<script>` tags directly in our HTML, but we can also write in separate .js files and link them in using `<script>` tags.
  - Consider advantages and disadvantages of these techniques.

Week 10

- Writing JavaScript
  - JavaScript code can go inside of `<script>` tags directly in our HTML, but we can also write in separate .js files and link them in using `<script>` tags.
  - Consider advantages and disadvantages of these techniques.

- Document Object Model
  - Recall from Week 6 that an HTML page structure allows for a website to be viewed hierarchically as a tree.
  - JavaScript provides an object, the *document object*, that we can manipulate to change our site and retrieve data on the fly.

Week 10

- Ajax
  - Formerly stood for *Asynchronous JavaScript and XML*.
  - Now we use JSON instead of XML, but this technology allows us to submit additional/supplementary HTTP requests without refreshing our websites.
  - Usually used to create a more enjoyable UX.

Week 10

- Ajax
  - Formerly stood for *Asynchronous JavaScript and XML*.
  - Now we use JSON instead of XML, but this technology allows us to submit additional/supplementary HTTP requests without refreshing our websites.
  - Usually used to create a more enjoyable UX.

- jQuery
  - jQuery is a popular library for JavaScript; it is not a language itself.
  - Extremely useful for DOM manipulation as well as handling Ajax requests, as it is cross-platform, abstracting away browser idiosyncracies.

Week 10

- APIs
  - *Application Programming Interfaces*
  - Suites of functions that handle the lower-level details for you, adhering to a promise for how a function should operate, so you can rely on that functionality in your own programs.
  - Bootstrap, Google, Facebook, and so much more (including simple APIs like ones written by CS50 or others).
  - Stand on the shoulders of those who came before, and rely on the work of others to develop applications of your own.