CS50 Quiz Review

November 13, 2017

Info

http://docs.cs50.net/2017/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 11/14 at noon, due via submit50 Thu 11/16 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/2017/fall/quiz/about.html

- Any topic that we have covered in the entire course is fair game.

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

Resources

- Consult the syllabus for a guide of topics.
- Review lecture notes.
- Review lecture source code.
- Review lecture slides.
- (Re)watch lecture videos.
- Review problem set specifications, distribution code, and sample solutions.

Resources

- Office hours
  - Tonight (here!) from 8pm – 10pm.

Resources

- Office hours
  - Tonight (here!) from 8pm – 10pm.

- No office hours during the Test (11/14 through 11/16) or indeed for the remainder of the semester!

Resources

- CS50 Discourse
  - You may post questions through the end of the day today.
  - You may <u>not</u> post questions on Discourse from Tue 11/14 through Thu 11/16.
    - Staff will not respond to any questions during this time, but will monitor the forum.

Resources

- CS50 Discourse
  - You may post questions through the end of the day today.
  - You may not post questions on Discourse from Tue 11/14 through Thu 11/16.
    - 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 Test are the course's heads.
- The course heads will answer only logistical and clarification questions that arise during the test.

Resources

- Quiz, Fall 2016
  - Available at https://docs.cs50.net/2017/fall/quiz/about.html, along with an answer key.

- Test, Fall 2016 and Fall 2017
  - 2017’s still available at http://cdn.cs50.net/2017/fall/test/test.html, along with an answer key (see David’s email from after the Test was finished).
  - 2016’s available at https://cdn.cs50.net/2016/fall/test/test.html

Resources

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

Weeks 0-5

- For all this material, consult the Test review session video and slides, available at https://cs50.harvard.edu/weeks under Week 6

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

Week 6

- Dynamic Programming
  - Saving results of computation to be used later
    - This process is called memoization
    - Results often stored in a “lookup table”
  - Often, larger problems can be solved recursively in terms of the solutions to smaller problems
  - Can often make algorithms more efficient

Week 6, continued

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

Week 6, continued

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

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

Week 6, continued

- 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 6, continued

- 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 6, continued

- 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 (almost entirely 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 7

- 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 associate functions with routes.
        - How to render—generate—page templates in HTML.
        - How to handle page redirection.
    - Uses Python *decorators* to associate the behavior of a method with a route (URL) that the user of the site might visit.

Week 8

- SQL
  - *Structured Query Language*
  - A language that is used for making requests, also known as *statements*, to databases.
  - Unlike HTML and CSS, is a programming language.
  - SQL is <u>not</u> the database itself.
    - MySQL, PostgreSQL, SQLite, MS 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 8

- SQL Syntax
  - `SELECT col1, col2 FROM table_name WHERE condition`
  - `INSERT INTO table_name (col1, col2) VALUES(val1, val2)`
  - `UPDATE table_name SET col1 = val1 WHERE condition`
  - `DELETE FROM table_name WHERE condition`
- Other Syntax
  - `PRIMARY KEY`
  - `UNIQUE`
  - `NOT NULL`
  - `AUTOINCREMENT`
  - `FOREIGN KEY`

Week 8

- 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 8

- 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.
- SQL Query for JOINing Tables

  -
    ```
    SELECT * FROM Album JOIN Artist ON Album.ArtistId = Artist.ArtistId
    ```

Week 8

- Injection attacks
  - Remember that buffer overflow attacks exploit the nature of the stack in C to perhaps redirect user’s code.
  - Injection attacks exploit the nature of SQL syntax to try to trick the database into executing a query the programmer did not intend to execute.
  - Defend against this by *escaping* user input to prevent dangerous SQL queries, which cs50.SQL.execute does for you.

Week 9

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

Week 9

- 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 9

- 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 9

- 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 9

- 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.
  - Never truly necessary, but used to create a more enjoyable UX.

Week 9

- 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.
  - Never truly necessary, but 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 9

- 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.
  - Yahoo, 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.