



T.P.S. REPORT

COVER SHEET

Quiz 1

out of 93 points

Print your name on the line below.

Do not turn this page over until told by the staff to do so.

This quiz is "closed-book." However, you may utilize during the quiz one two-sided page (8.5" × 11") of notes, typed or written, and a pen or pencil, nothing else.

Scrap paper is included at this document's end.

Circle your teaching fellow's name.

Alex Chang
Alex Hugon
Ana Roda
Ashin Shah
Ben Massenburg
Dan Armendariz
Dev Purkayastha
Doug Lloyd
Ellen Farber
Ferris Zhang
Fil Zembowicz
Gabrielle Ehrlich
Idriss Fofana
Jeff Solnet
Jeremy Cushman
John Tristan
Josh Bolduc

Katie Fifer
Ken Parreno
Lakshmi Parthasarathy
Lauren Carvalho
Lee Evangelakos
Luis Duarte
Marta Bralic
Matt Chartier



Melissa Niu
Michael Chen
Michael Oberst
Mike Teodorescu
Peter Hung
Punit Shah
Rei Diaz
Rob Bowden
Rose Cao
Scott Crouch
Sophie Chang
Steve Tricanowicz
Tian Feng
Tommy MacWilliam
Wellie Chao
Willie Yao
Yuhki Yamashita

for staff use only

final score out of 93

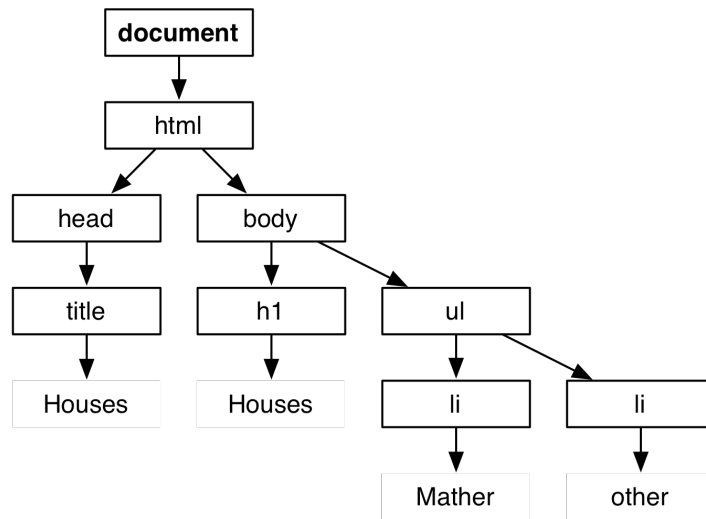
True or False.

For each of the statements below, circle T if the statement is true or F if the statement is false.

0. T F (0 points.) You should use a `string` to tie balloons together.
1. T F (1 point.) Web servers generally listen for requests on port 80.
2. T F (1 point.) Both JavaScript and PHP are interpreted languages.
3. T F (1 point.) PHP supports associative arrays.

DOM DOM DOM.

Consider the DOM below whose leaves are text nodes.



4. (3 points.) Beneath the `DOCTYPE` below, write the HTML that corresponds to this DOM.

`<!DOCTYPE html>`

for staff use only

—

Bah.

Suppose that you're at dinner with some friends who've not (yet) taken CS50. And so they have some misconceptions about Firesheep. Respond to your friends' comments below, explaining in a sentence or two why they are not, in fact, as safe from session hijacking as they think.

5. (2 points.) Oh, I'm safe, I don't use Facebook.

6. (2 points.) Oh, I'm safe, the last session cookie I received from a website was 212 alphanumeric characters long. There's no way some Firefox plugin can guess that.

7. (2 points.) Oh, I'm safe, every website I visit uses SSL when you log in, so my username, password, and cookies are encrypted.

8. (2 points.) Oh, I'm safe, I've started VPNing to Harvard's VPN server anytime I'm on wireless so that no one can hijack my sessions.

for staff use only

—

Frosh IMs.

Consider the HTML below.

```
<form action="http://froshims.net/register.php" method="get" name="register">
  Name: <input id="name" name="name" type="text">
  <br>
  Gender:
  <input name="gender" type="radio" value="F"> F
  <input name="gender" type="radio" value="M"> M
  <br>
  House:
  <select name="dorm" size="1">
    <option value=""></option>
    <option value="matthews">Matthews Hall</option>
    <option value="other">Other</option>
  </select>
  <br>
  <input type="submit" value="Register!">
</form>
```

9. (2 points.) Suppose that David, a guy from Matthews Hall, fills out this form completely. At what URL will he find himself after submitting the form?
10. (3 points.) Suppose that Matthews Hall is tired of losing and decides that it is time to ban David from registering for sports. And so

```
onsubmit="return validate();"

```

is added to the HTML's `form` element. Complete the implementation of `validate` below in such a way that anyone named David will be unable to submit the registration form. No need for any alerts; it suffices to prevent the form's submission for anyone named David.

```
function validate()
{

```

for staff use only

—

Shuttletime.

Suppose that a shuttle with 35 seats is implemented in C as a global array whose size is declared with a constant, per the below.

```
#define SEATS 35
bool seats[SEATS];
```

Suppose that the shuttle's seats are initially flagged as empty (*i.e.*, `false`), per the initialization below.

```
for (int i = 0; i < SEATS; i++)
    seats[i] = false;
```

11. (2 points.) Complete the implementation of `taken`, below, in such a way that the function returns `true` if the specified seat (*i.e.*, `bool`) is already taken (*i.e.*, `true`) else `false` if the specified seat is out of bounds or empty.

```
bool
taken(int seat)
{
```

12. (2 points.) Complete the implementation of `take`, below, in such a way that the function flags the specified seat (*i.e.*, `bool`) as taken (*i.e.*, `true`) and then returns `true`, unless the specified seat is out of bounds or already taken, in which case the function should instead return `false`.

```
bool
take(int seat)
{
```

for staff use only

—

Shuttletime, continued.

13. (1 point.) A `bool` is generally implemented with 8 bits, since a byte is the smallest unit that most CPUs can address, even though 1 bit should suffice to represent two possible values like `true` and `false`. And so, at the moment, the global array, `seats`, consumes 35 bytes instead of 35 bits, which means that $35 \times 8 - 35 = 245$ bits are essentially wasted. Suppose that we instead want to represent each seat with 1 bit, and so we re-declare `seats` as having just enough bytes for that many bits, per the below.

```
uint8_t seats[BYTES];
```

The implication is that `seats[i]` no longer represents 1 seat but 8 seats. If the shuttle still has 35 seats in total, what should the value of `BYTES` be?

14. (1 point.) Of course, 35 is not a multiple of 8, so how many bits are now wasted?
15. (2 points.) We must still flag all seats as initially empty. Complete the implementation of the `for` loop below in such a way that it sets all bits in `seats` (including any wasted) to 0s.

```
for (
```

16. (6 points.) We must now rewrite `taken`, since `seats[i]` no longer represents 1 seat but 8 seats. Complete the implementation of `taken`, below, in such a way that the function returns `true` if the specified seat (*i.e.*, bit) is already taken (*i.e.*, 1) else `false` if the specified seat is out of bounds or empty. You may assume that both `SEATS` and `BYTES` are available to you as constants.

```
bool  
taken(int seat)  
{
```

for staff use only

—

17. (6 points.) We must also rewrite `take` because `seats[i]` no longer represents 1 seat but 8 seats. Complete the implementation of `take`, below, in such a way that the function flags the specified seat (*i.e.*, bit) as taken (*i.e.*, 1) and then returns `true`, unless the specified seat is out of bounds or already taken, in which case the function should instead return `false`. You may assume that both `SEATS` and `BYTES` are available to you as constants.

```
bool
take(int seat)
{
```

Pointer Fun, still without Binky.

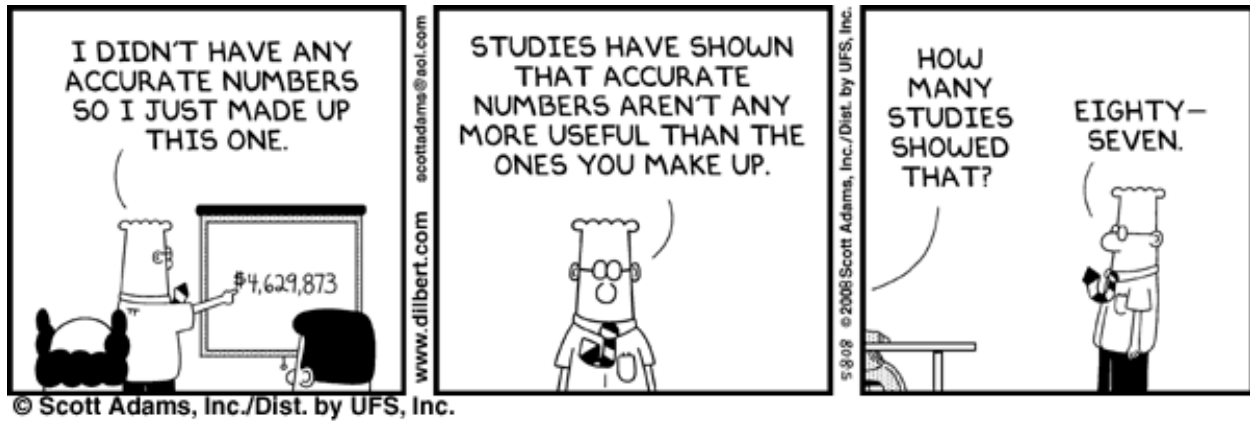
18. (1 point.) Consider the function below.

```
void
f(char *s)
{
    if (s != NULL)
    {
        for (int i = 0, n = strlen(s); i < n; i++)
            printf("%c\n", *(s+i));
    }
}
```

State in a sentence precisely what this function does.

for staff use only

—



Numeric Self Defense.

For each of the claims below, state whether the statistic is much too high, much too low, or plausible, and then argue why in no more than three sentences. Be sure to back your arguments with some arithmetic, even if your numbers are estimates.

19. (2 points.) Back when David took CS50 in 1996, his laptop had only 4MB of RAM. Thanks to technological advances, the amount of RAM in laptops has doubled every 18 months ever since.

20. (2 points.) The number of students in CS50 has doubled nearly every year since the course's debut in 1989.

for staff use only

—

Design Decisions.

For each pair below, x versus y , argue in no more than two sentences when you should use x rather than y (or, if you prefer, y rather than x).

21. (2 points.) *JavaScript* versus *PHP*
22. (2 points.) *passing by reference* versus *passing by value*
23. (2 points.) `gdb` versus `valgrind`
24. (2 points.) `long` versus `int64_t`
25. (2 points.) *GET* versus *POST*

for staff use only

Bases Covered.

26. (4 points.) Complete the table below in such a way that each row's values are equal. It's fine to omit leading zeroes.

Binary	Decimal	Hexadecimal
00000000	0	0x00
00000010	2	
	10	
00010000		0x10

Structures.

Consider the pseudocode below, where *s* represents some data structure that's initially empty. Assume that *push* adds a number to *s* and that *pop* removes a number from *s*.

```
s.push(1);
print(s.pop());
s.push(2);
s.push(3);
print(s.pop());
s.push(3);
print(s.pop());
s.push(4);
s.push(5);
s.push(6);
s.push(7);
print(s.pop());
```

27. (2 points.) If *s* is a queue, what gets printed?

28. (2 points.) If *s* is a stack, what gets printed?

for staff use only

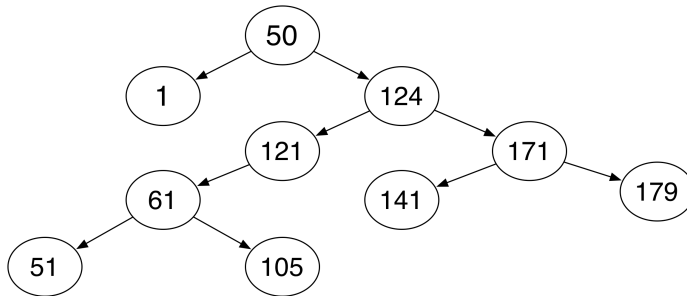
—

More Structures.

Suppose that a hash table for (entirely alphabetical) English words is implemented as an array with 26 separate chains.

29. (2 points.) Even with an optimal hash function that uniformly distributes words over the 26 chains, searching a hash table is in $O(n/26)$, which is equivalent to $O(n)$ since 26 is a constant. And so a hash table's efficiency seems no better than one big linked list. In a sentence or two, why bother using a hash table at all, then, for English words?

30. (6 points.) Recall that that value in any node of a binary search tree (BST) is greater than that of its left child, if any, and any descendants thereof as well as less than that of its right child, if any, and any descendants thereof, as in the example below.



Suppose that each node in this tree is of type `node`, per the below, whereby `left` and `right` are initialized to `NULL` before any insertions.

```
typedef struct node
{
    int n;
    struct node *left;
    struct node *right;
}
node;
```

Complete the implementation of `print_r` on the following page in such a way that the function traverses the given BST recursively, ultimately printing nodes' values in order, from smallest to largest, with one comma between each pair of values. For example, in the case of the BST above, this function would print the below.

for staff use only

1, 50, 51, 61, 105, 121, 124, 141, 171, 179

Assume that `tree` is always a pointer to a BST or `NULL`. As always, take care not to segfault.

```
void  
print_r(node *tree)  
{
```

Axe to Valgrind.

31. (1 point.) Suppose that `valgrind` reports the error below when run on Cansu's code. Explain in a sentence what Cansu has probably done.

```
Invalid write of size 4
```

32. (1 point.) Suppose that `valgrind` reports the error below when run on Yuhki's code. Explain in a sentence what Yuhki has probably done.

```
40 bytes in 1 blocks are definitely lost in record 1 of 1
```

for staff use only

—

Quickies.

Answer each of the questions below in no more than 3 sentences.

33. (1 point.) What's steganography?

34. (1 point.) Why bother with external CSS files (e.g., `styles.css`) when you can just add a `style` attribute to HTML elements?

35. (2 points.) What's two-factor authentication?

36. (1 point.) Why are `XMLHttpRequest` objects useful?

37. (2 points.) What's the difference between `NULL` and `'\0'`?

38. (1 point.) What's an HTTP cookie?

for staff use only

—

39. (1 point.) What's the relevance of the so-called Birthday Problem to hash tables?

40. (1 point.) What's an event handler?

Hello, Katie.

41. (6 points.) Suppose that Katie's goal is, quite simply, to print `hello` to the screen, but she doesn't recall how to in C, and so she's written three versions, below. Explain next to each version whether it is correct or incorrect. If incorrect, explain what will actually happen when the code (if written as part of a program with no other mistakes) is compiled and executed.

```
// version 1
char *s = "hello";
printf("%s", *s);
```

```
// version 2
char *s = "hello";
printf("%s", s);
```

```
// version 3
char *s = "hello";
printf("%s", &s);
```

for staff use only

—

My oh my, SQL.

Consider the schema below, which describes the SQL table that your broker uses to keep track of clients. Assume that the table is called `clients`.

Field	Type	Collation	Attributes	Null	Default	Extra
id	int(10)		UNSIGNED	No	None	AUTO_INCREMENT
username	varchar(255)	latin1_swedish_ci		No	None	
password	varchar(255)	latin1_swedish_ci		No	None	
cash	decimal(65,4)		UNSIGNED	No	0.0000	

42. (1 point.) Suppose that your broker has decided to impose a minimum balance fee, whereby clients who don't maintain a cash balance of at least \$5,000 are charged \$20 per month. What SQL query should your broker execute on this table each month in order to deduct \$20 from every such client's cash balance? You may assume that every such client will always have at least \$20 in cash.
43. (1 point.) Suppose that a client has been accused of insider trading, and so it is time to delete his account. What SQL query should your broker execute on this table in order to delete the client whose username is `dshen`? No need to withdraw the client's cash first.
44. (3 points.) Suppose that your broker wrote the following PHP code to handle logins.

```
$username = $_POST["username"];
$password = $_POST["password"];
$sql = "SELECT id FROM clients WHERE username='$username' AND password='$password'";
$result = mysql_query($sql);
if (mysql_num_rows($result) != 0)
{
    $row = mysql_fetch_array($result);
    $_SESSION["id"] = $row["id"];
}
```

Suppose, though, that a hacker inputs the below as his or her password when prompted to log in.

`' OR '1' = '1`

Explain in one or more sentences exactly what will happen and why. And propose in a sentence how to fix.

for staff use only

—

Scrap Paper.

Nothing on this page will be examined by the staff unless otherwise directed in the space provided for some question.