

CS193i Final Exam

SITN Students: ideally this exam should get back to me via courier or fax by Friday the 12th so there is some chance of finishing the grades by the deadline. Vicky Carrillo is the logistical contact — carrillo@leland.stanford.edu, 650 725-3015.

Welcome to the CS193i Final Exam. The exam is 2 hours long and is open book, open note, but closed to computer access. There are 5 problems for 100 points total and your exam should have 11 pages.

For shallow, irrelevant errors such as language syntax or mixed up variables, we will try to interpret what you meant. We will try to grade on the core course concepts: sockets, the CGI dialog, inheritance in Java...Strategically, try not to get bogged down on a single problem— there is not enough time to look up too many things. Move on to questions you can do quickly and come back if you have time. No problem should take more than 20 or 30 minutes. Good luck and have a nice summer!

Name (printed) _____

I accept the letter and spirit of the honor code— I have not given or received aid on this exam.

(signed) _____

	Score	Grader
problem 1 (25)	_____	_____
problem 2 (25)	_____	_____
problem 3 (20)	_____	_____
problem 4 (15)	_____	_____
problem 5 (15)	_____	_____
Total (100)	_____	

Today's quote: The cost of feathers has risen.... Now even down is up!

1) TCP/IP and HTTP (25 points)

First, here is some reminder code for the socket interfaces you may want to consult for this problem. You do not need to check the error returns for reading and writing — just for hostname lookup and connecting.

C Socket Code

```
struct hostent *hostInfo;
struct sockaddr_in inetAddr;
int sock;

inetAddr.sin_family = AF_INET;
inetAddr.sin_port = <port num>;
hostInfo = gethostbyname(<hostname>); // returns NULL on err
memcpy(&inetAddr.sin_addr, hostInfo->h_addr, hostInfo->h_length);
sock = socket(PF_INET, SOCK_STREAM, 0);
connect(sock, &inetAddr, sizeof(inetAddr)); // returns -1 on err

once the socket is setup, you can read and write with...

int write(int sock, char* buff, int buffLen); // you may ignore the return value
int read(int sock, char* buff, int buffLen); // returns 0 on EOF
```

Perl Socket Code

```
$ipaddr = inet_aton(<hostname>); # returns false on error
$sockaddr = sockaddr_in(<port num> , $ipaddr);
socket(SOCK, PF_INET, SOCK_STREAM, 0);
connect(SOCK, $sockaddr); # returns false on error

Once connected, the socket can be read with <SOCK> and written to with:
print SOCK <string>;
```

For this problem, you need to write a small program which retrieves a series of web pages. Standard input (STDIN) to the program will consist of a series of absolute (not relative) HTTP URL with the leading "http://" omitted. Each URL ends with a complete path+file beginning with a single '/'. For simplicity, the path+file will not be the empty string and will not have a suffix. So the input will look like...

```
cse.stanford.edu/class/
www.whitehouse.gov/pets/socks.html
www.yahoo.com/
...
```

The program should print each URL to standard output and then try to do a single HTTP GET for the URL. If the hostname is bad or the connection fails, then print the error message "ERROR hostname" or "ERROR connection" after the URL with no further processing required for that URL. These are the only two error cases which the program must consider. Otherwise, if the connection is successful, the program should read all of the header and body of the HTTP response and simply print it all out as returned by the server. In any case, the program should then continue processing URLs until the input is exhausted. You may build your solution in C or Perl (just as on the homework), but you should not use any HTTP specialized libraries other than Socket. You do not need to worry about decomposition or style of your solution — we will grade only on the proper functioning of your code.

(Solution to TCP/IP problem)

2) CGI (25 points)

For this problem, you will build a simple CGI which supports append-only files of text on the server which can be used to gather people's favorite quotations. There are three quotation files -- "love", "war", and "scrabble" — each is stored as a text file of the same name in the directory with the CGI. The start URL for the CGI is <http://www.stanford.edu/cgi-bin/quote.pl>

No error checking, concurrency safety, or security is required for this problem.

a) When invoked with no bindings, the CGI should return an HTML page which contains a simple `` for each of the three files. Each href should return the content page ((b) below) for that file.

b) The content page for each file is just an HTML page titled with the name of the file and including the entire contents of the file. At the bottom of the page (after the contents of the file) there should be a simple form which allows the user to append new text to the file. The simple form can show just a single text input field and a single submit button. Submitting the form should...

c) Submitting the form should append the new input to the end of the file. This may be done by reading in the entire file and then writing it out again with the new data (this is how WebDB worked). Alternately, in Perl, opening a file with the command `open(F, ">>${fname}")` opens a filehandle F for "append" mode — printing to the file handle automatically appends to the file. The submit should return a simple "Ok, it worked" page. You may write your solution starting with the skeleton of the CGI code below. This starter code includes the basic call to the 193icgi.pl library, but nothing else...

```
#!/usr/bin/perl
require "193icgi.pl";

%pairs = ExtractPairs();
```

More room for the CGI problem...

3) The Bill Zoo Applet (20 points)

This simple Java Applet will contain two components: a Zoo canvas and a button. You should read all four parts of the question first, since ideally the parts depend on each other a bit (you can get credit for a part without having completed the others however).

a) Write the ZooApplet code so it contains a single Zoo canvas and below that a "Move" button which initiates a move() message to the Zoo.

b) The Zoo is a simple 200 by 200 canvas — all it does is represent an area containing single "Bill" animal which is at some x,y location in the canvas at any time. Initially, the Bill is at a random location. On screen, the Zoo should appear only as the string "Bill" at Bill's current location (use `g.drawString(x,y,"Bill")`). The Bill should not be a separate object — it is really just represented by the current Bill x,y values stored in the Zoo. Zoo should respond to a `move()` message which moves Bill to another random location. You may assume that Zoo already has the same random number helper function as the Account example:

```
// Returns a random int in the range [0..upperBound-1]
private int randomNum(int upperBound)
```

Write the Zoo class (except you can just assume that `randomNum()` is defined for you)...

c) Write a Mover Thread subclass which can take care of the moving the Bill around a little automatically. The Mover should send a move() message to the Zoo and then sleep(10000) to wait 10 seconds. You may omit the try...catch block.

d) Show the code which would need to be added to the end of the Applet init() to create a Mover so that Bill would move around a little on his own wether the button was pressed or not.

4) OOP Design (15 points)

Java Inheritance

Suppose there is a Java program which includes a Person class which simulates daily eating behavior. The Person class includes the following "oneDay" method which runs the receiver through the three meals of the day.

```
// Person class...
public void oneDay() {
    breakfast();
    lunch();
    dinner();
}
```

The three methods breakfast(), lunch(), and dinner() have non-abstract definitions in the Person class which implement the standard behavior for each meal.

Now suppose that there are three variations which eat more than the standard Person...

B people are like Person, but at breakfast time, they have two breakfasts in succession.

L people are like Person, but at lunch time, they have two lunches in succession.

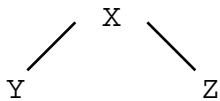
D people are like Person, but at dinner time, they have two dinners in succession.

a) Suppose the Person class has already been defined — what is the best way to define the B, L, and D classes in way that all the objects will respond to the oneDay() message properly. The following drawing shows the Person class with the methods it defines. Draw in the other classes for your design showing their relationship to Person (as I did in lecture) and show what methods they implement (no code is necessary at this step)...

```
Person
  -oneDay()
  -breakfast()
  -lunch()
  -dinner()
```

b) Consistent with your (a) design drawing, give the code for just the B class...

c) Suppose X is the superclass of Y and Z...



Consider the following code. The first three declarations are fine and then there is a single assignment line for which there are four possible choices. Which, if any, of the choices will lead to compile-time or run-time errors and what is the error in each case?. Note that each variable is named for its run-time type, but the compile time types are not that simple.

```

{
  X x = new X;
  X y = new Y;
  Z z = new Z;

  <the assignment line goes here>
}

// the choices for the assignment line...

x=y; // choice 1

y=x; // choice 2

x=z; // choice 3

z=x; // choice 4

```

5) Short Answer (15 points)

a) When an HTTP server responds to a valid GET request, what are the two parts of the response, what separates them, and how does the client know what type of data is being returned?

b) Suppose the code for a CGI looks like the following...

```
// $fname is set from the bindings in the $QUERY_STRING
open(F, "$fname");
// read and process data from F
```

This CGI code has a security hole. What is an example of something a malicious client could do to exploit this hole?

c) Suppose the following HTML occurs in a document whose URL is `http://foo.com/dir/a.html`

Now suppose that each of the hrefs is clicked resulting in a "GET XXX HTTP/1.0\r\n\r\n" request being sent to the appropriate server. Indicate what exactly XXX will be to the right of each href...

```
<a href=binky/b.html>1</a>
<a href="c.html#foo">2</a>
<a href=/baz>3</a>
<a href=http://cse.stanford.edu/class/>4</a>
<a href="http://oof.com/cg-bin/echo.cgi?num=42">5</a>
```

d) True or False: HTML should produce the same appearance no matter what computer is displaying it.

e) When a computer is physically moved from one network to another, its TCP/IP address may need to be changed. Why?