

**PRACTICE QUESTIONS FOR 15.561 FINAL EXAMINATION  
Spring 2005**

**CLARIFICATION: This is not a practice final but a collection of questions similar to those likely to be on the final.**

**COMPUTER FUNDAMENTALS PRACTICE QUESTIONS**

**Multiple-choice questions: SELECT THE BEST ANSWER.**

- C 1. All of the following can fit in one byte, EXCEPT:
- A. A letter encoded in ASCII.
  - B. One out of 200 different possible machine instructions.
  - C. A memory address for a computer which has 64KB RAM.
  - D. 4 pixels where each pixel can be either black, white, red, or blue.
  - E. One pixel where each pixel can have one of 128 different colors.
- B 2. A modern microprocessor can process information at a rate that is:
- A. slower than a hard disk's capability to store the data.
  - B. faster than a hard disk's capability to store the data.
  - C. comparable with a hard disk's capability to store the data.
  - D. faster than the transfer rate of a hard disk, but slower than that of a CD-ROM.
  - E. faster than the transfer rate of a CD-ROM, but slower than that of a hard disk.
- C 4. The amount of video memory required to store a screen with a resolution of 1024x1024, where each pixel can have one of 65536 different colors, is:
- A. 500KB
  - B. 1MB
  - C. 2MB
  - D. 3MB
  - E. 4MB

- D\_\_ 5. All of the following influence the storage capacity of a magnetic hard disk, EXCEPT:
- A. The number of tracks per recording surface.
  - B. The bit density of the recording technology.
  - C. The number of recording surfaces.
  - D. The speed of rotation of the recording surface.
  - E. The diameter of the recording surfaces.
- D\_\_ 6. All of the following statements describe instances of the general technique of caching, EXCEPT:
- A. Pentium processors store copies of the most frequently accessed memory locations internally (on-chip), so that subsequent data accesses can be satisfied very quickly.
  - B. Hard disk drivers store copies of the most recently accessed hard disk clusters in RAM buffers.
  - C. Netscape stores copies of the most recently accessed Web pages in a special directory of the local hard disk.
  - D. Windows 98 allows users to place icons, representing shortcuts to their most frequently accessed applications, on the Windows desktop.
  - E. CD-ROM drivers store copies of the most recently accessed CD-ROM blocks on hard disk.

## COMPUTER NETWORKS PRACTICE QUESTIONS

**Question 1:** Digital technologies are rapidly replacing analog technologies in applications ranging from data communication to telephony, television, audio reproduction etc. List **three** advantages of digitally encoded data relative to data encoded in analog form.

1. Digital signals are more resilient to distortion and interference than analog signals. Using digital signals, every data value is transmitted using a discrete signal "type". Therefore, even when a signal gets somewhat distorted during transmission, the receiver can usually still correctly decide which of the allowed signal "types" is "closest" to each received pulse and thus reconstruct the correct data.
2. Digital signals can be multiplexed using time division multiplexing, which results in better bandwidth utilization than frequency division multiplexing (In frequency division multiplexing, multiplexed signals must be separated by unused frequency zones, or "gaps". This reduces the total bandwidth utilization).
3. Digital signals can be compressed and packetized to achieve faster effective transmission rate.
4. Digital signals can be encrypted in applications that require high levels of security.
5. Applications using digital data can take advantage of the remarkable improvements in cost/performance described by Moore's Law.

**Question 2:** Consider the following simple analogy used to describe how the Internet works:

"Imagine a complex highway system on which millions of cars are always moving. There are groups of cars that belong to the same travel party heading to the same place. Before getting into the highway, each individual driver consults a map and marks an itinerary from origin to final destination. Each driver is doing this independently. Therefore, it is possible that different cars belonging to the same travel party may follow different itineraries. Eventually, however, a car arrives at its destination. It waits for other members of the same travel party to arrive, and then they all do something useful together."

Compared with the way the Internet actually works, the above analogy contains an important inaccuracy. Can you explain what it is?

The inaccuracy is that each car (packet) contains a preplanned itinerary (route through the network) from origin to destination.

Packet-switching networks, such as the Internet, do not plan a route for each packet in advance. Each packet includes the IP address of its final destination only. Based on this information, each router in the network first checks to see if it has a direct point-to-point connection with the packet's destination. If it has then it sends it there. Otherwise it forwards the packet to the "best" router it has a direct point-to-point connection with. The "best" router is the one, which is "closest" to the packet's final destination. The process is repeated until the packet reaches its final destination.

## SOFTWARE DEVELOPMENT PRACTICE QUESTIONS

**Question 1:** There are tradeoffs between programming in a high-level language vs. assembly language. Compare these by giving at least one advantage of working at each level.

Using a high-level language is more user-friendly. It is conceptually more natural, and usually easier to specify what you want the computer to do. The higher level hides more details of how things work at lower levels; the human programmer doesn't have to know so much about the computer itself (i.e., its lower levels) and can achieve a task (e.g., specify how to do it) with less labor time. A disadvantage of working at the higher level is that more computation (at execution time) is usually required by the computer.

Programming in assembly language gives more control over the details of what the machine actually does - if one has the patience and expertise to properly exploit this control. This means that you can often write assembly language programs that require less running time and/or less memory space than the equivalent programs written in a higher level language would require. Working in assembly language also facilitates integration with other software that is lower-level.

**Question 2:** For each of the following approaches to developing software, please list one advantage and one disadvantage of the approach:

Prototyping

**•Advantages**

- Especially useful when exact requirements are hard to know in advance
- »user interfaces
- »decision systems
- »electronic commerce?
- Encourages user involvement

**•Disadvantages**

- Hard to predict and control outcomes reliably
- If repeated, significant reimplementations are needed, can be very expensive
- May result in systems that are inefficient, unreliable, or hard to maintain

Packaged software

**•Advantages**

- By amortizing development and maintenance costs over many organizations, it is possible to get superior solutions at much lower cost

**•Disadvantages**

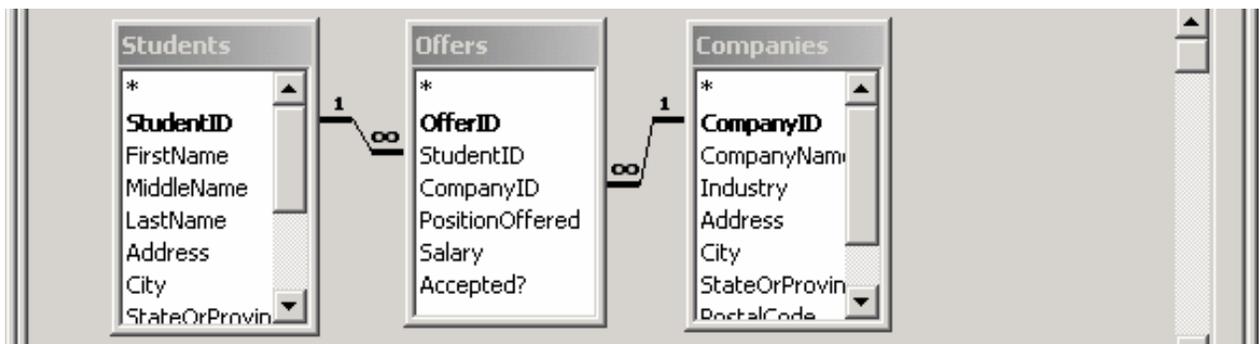
- Customizing software can be very time-consuming and expensive
- May have to change organization to fit software, rather than vice versa

## DATABASE DESIGN PRACTICE QUESTION

The Sloan Career Development Office has hired you as a consultant to help design a system using a relational database to maintain placement statistics for graduating MBA students. The database schema (tables and fields) has already been designed by someone else (see below). Your job is to define a set of queries for the system.

As you specify these queries on the following page, please bear in mind the following:

1. For each query, please fill in the spaces as you would in the Microsoft Access Query Design View.
2. If your query does *not* need to use the row labeled “Total”, please *cross out* that row.
3. If your query *does* need to use the row labeled “Total”, you need to fill in something in that row for each column you use. The possible values you can use are:
  - a. Group by
  - b. Sum
  - c. Avg
  - d. Min
  - e. Max
  - f. Count
  - g. StDev
  - h. Var
  - i. First
  - j. Last
  - k. Expression
  - l. Where
4. If your query needs something in the row labeled “Sort”, you can use either of the two following values:
  - a. Ascending
  - b. Descending
5. In the row labeled “Show”, you can either leave the box blank or put a “ ” in it.
6. Use the following data schema for all the queries in this section:



1. Create an alphabetical list of students, together with the total number of offers each student received.

Field:				
Table:				
Total:				
Sort:				
Show:	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Criteria:				
or:				

2. Create an alphabetical list of the students who received offers from the company named "McKinsey".

Field:				
Table:				
Total:				
Sort:				
Show:	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Criteria:				
or:				

3. Create a table showing all companies that made offers in order of the *average* salary they offered. (Companies that offered the highest average salaries should be listed first.)

Field:				
Table:				
Total:				
Sort:				
Show:	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Criteria:				
or:				

4. Create a table showing all companies whose offers were accepted, in order of the number of accepted offers. (Companies with the most accepted offers should be listed first.) You can assume that the field "Accepted?" has the value "Yes" for accepted offers.

Field:				
Table:				
Total:				
Sort:				
Show:	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Criteria:				
or:				

**Answer Key for Database Design Question:**

1. Create an alphabetical list of students, together with the total number of offers each student received.

Field:	StudentID	LastName	FirstName	OfferID		
Table:	Students	Students	Students	Offers		
Total:	Group By	Group By	Group By	Count		
Sort:		Ascending	Ascending			
Show:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Criteria:						
or:						

*Note: The first column in this query (“StudentID”) is included to account for the (rare) cases where two or more students have the same first and last names. Your answer would be counted as correct even if you did not include this column. A similar comment applies to the other queries below.*

2. Create an alphabetical list of the students who received offers from the company named “McKinsey”.

Field:	StudentID	LastName	FirstName	CompanyName		
Table:	Students	Students	Students	Companies		
Sort:		Ascending	Ascending			
Show:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Criteria:				"McKinsey"		
or:						

*Note: The “Total” row is not shown in this screen image. On your exam sheet, you should cross it out.*

3. Create a table showing all companies that made offers in order of the *average* salary they offered. (Companies that offered the highest average salaries should be listed first.)

Field:	CompanyID	CompanyName	Salary			
Table:	Companies	Companies	Offers			
Total:	Group By	Group By	Avg			
Sort:			Descending			
Show:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Criteria:						
or:						

4. Create a table showing all companies whose offers were accepted, in order of the number of accepted offers. (Companies with the most accepted offers should be listed first.) You can assume that the field “Accepted?” has the value “Yes” for accepted offers.

Field:	CompanyID	CompanyName	OfferID	Accepted?		
Table:	Companies	Companies	Offers	Offers		
Total:	Group By	Group By	Count	Where		
Sort:			Descending			
Show:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Criteria:				"Yes"		
or:						

## INTERNET SECURITY PRACTICE QUESTION

An increasing number of retail banks are offering Internet banking services. Using those services, bank customers are able to access their accounts and perform transactions through a Web browser. Internet banking introduces a whole set of security concerns, most of which are not present in ATM networks. More specifically, the following concerns are present:

- **Concern 1:** Anybody can connect to the bank's Web site and try to access an account. The bank must verify that it is transacting with the legitimate account owners
- **Concern 2:** Clever impostors can "impersonate" a bank's Web server and fool customers into supplying their account numbers and passwords. Before they send any sensitive information, bank customers need to verify that the Web site they are transacting with actually belongs to the bank
- **Concern 3:** The Internet is an insecure, public network. All traffic can be intercepted by unauthorized third parties. The bank must make sure that account numbers, passwords, account balances and other sensitive information cannot be revealed to third parties, even though network traffic might be intercepted.

Sketch a protocol that addresses all three security concerns listed above and allows a bank client and a bank server to communicate securely over the Internet. **In your solution you can assume that all necessary keys have already been securely transmitted to the parties that need to use them.**

Write up your answer as follows:

- List each step of your protocol in sequence.
- For each step of the protocol, list the sender and recipient of the message (use the abbreviations B=Bank, U=User) followed by a brief description of the message contents
- If a message is encrypted or signed, make sure you clearly state that and also list the key that was used to encrypt/sign the message.

For example:

1. U->B User sends her account number and deposit amount to the bank
2. B->U Bank performs operation and sends back the new account balance signed by B's private key.

Here's one possible solution:

1. U->B U sends the following message to B: "hello bank. I'm U. Here are the details of the transaction I want you to perform". U's message is encrypted by B's public key and signed by U's private key.
2. B->U B verifies U's signature and decrypts the message. It performs the requested transaction and responds to U by sending the following message: "This is the bank. These are the results of the transaction you requested". B's message is encrypted by U's public key and signed by B's private key.
3. U U verifies B's signature and decrypts the message. If all is well, it displays the transaction results to the human operator.

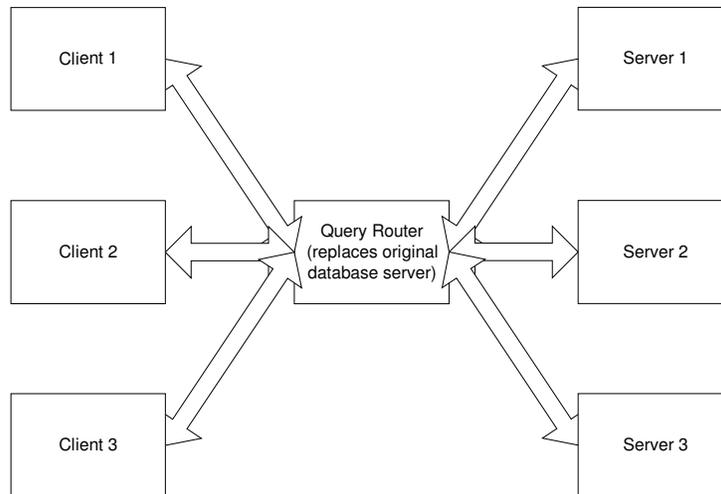
### **ENTERPRISE-LEVEL SYSTEMS PRACTICE QUESTIONS**

**(Note: You should expect a question of a third of the scope of the question below.)**

Question: A small bank is using a client/server system with a single database server for storing the transaction histories and account balances of all their customers. Users query the server by sending the account number and PIN of a customer, plus the type and parameters of the required transaction (e.g. deposit, withdrawal, transfer). The server performs the transaction and returns the new account balance of the customer. Recently, the bank has been experiencing rapid growth. As a result, the bank server has become overloaded. The bank has decided to reorganize its system in order to improve its performance by adding more servers.

To make the transition to the new system as smooth as possible, the bank does not wish to make any changes to the client portion of its software (the part which resides on every ATM and every bank teller's workstation and implements the user interface). However, the current client software has been designed to communicate with a single server only. The network address of the database server is hard-coded in the client software.

- a. Propose a high-level architecture for the new system. Draw a simple diagram of the new system and explain the role of all machines that are present in your diagram. To receive full credit, your design must not require any change to the client portion of the original system (15 points).



The basic idea is to convert the original, 2-tier client/server system into a 3-tier system by replacing the original database server with a "query router" machine. The query router receives queries from all clients (who "think" that they are still talking directly to the database server). The router simply forwards the query to the appropriate server (see answer to question c for how the appropriate server is determined). It then receives the response from the server and sends it back without any change to the client.

- b. Describe what the database of each new server should contain relative to the contents of the original server database (10 points).

Each database should contain a subset of the original accounts. In order to distribute the load evenly among the servers, ideally, all accounts should be partitioned evenly among the servers. For example, suppose that the range of account numbers is 1000000-8999999 and that there are two servers. Then, Server 1 should contain accounts 1000000-4999999 and Server 2 accounts 5000000-8999999

- c. Describe how client queries get distributed to the right server (5 points).

If we assume the above partitioning of accounts to servers, the query router only needs to look at the first digit of the account number supplied by the client in order to decide to which server it should be forwarded: All accounts that begin with 1-4 should be forwarded to Server 1, while all accounts that begin with 5-8 should be forwarded to Server 2.

- d. The above design was derived with the restriction that you could not modify the client part of the software. If you were free to modify the client part of the software, would you have designed the new system in any different way? If yes, then explain how and why. If not, then explain why not. No diagram is necessary, but you may draw one if it helps you explain your points faster (10 points).

The problem with the original system was that a centralized server was handling all the queries. The multi-server design proposed in question a. still has a centralized point, through which all queries must pass (query router). The query router does minimal processing on each query (see answer to question c) so it is expected to be much less of a bottleneck than the original centralized server. However, if the volume of queries increases significantly enough, the centralized query router will eventually become a bottleneck (especially because all the query results also pass through it). Finally, if the query router crashes, the entire system goes down with it.

One possibility would be to change the interaction protocol as follows: The client contacts the query router in order to get the network address of the appropriate server; the query router, based on the account number provides the network address of the appropriate server; the client then contacts the appropriate server independently.

This solution only involves the query router in order to locate the right server and is therefore much more scalable. If we were concerned with the possibility of the query router failing, we could improve our architecture even further by maintaining multiple, redundant query routers. Then, if a client failed to receive a response from one query router, it could try the others, one after another, until it got a response.

## **ENTERPRISE SYSTEMS AND WEB SERVICES PRACTICE QUESTIONS**

1. For the following technologies: (i) explain each briefly (one to two sentences), and (ii) list an advantage that each can provide to organizations.

- **ERP**

- **CRM**

- **SCM**

- **Ontology**

- **SOAP**
  
- **UDDI**
  
- **Web services**