1. **(a)** Good HCI design is based on values summarised by these five words: people, prototype, compare, iterate and principles. Briefly explain what each of these words refers to. In your answer, give a brief example of the practical application of each in HCI design.

[10]

Lecture material:

- *People refers to carrying out user studies and user trials so as to ground design in the real experience of people [1]. An example is giving a representative set of users relevant tasks to perform using a system, and measuring their performance [1].*
- *Prototype refers to making simplified versions of a system design so as to learn from their evaluation early on in the design process [1]. An example might be a mockup of an interface quickly made using paper or Powerpoint [1].*
- *Compare refers to the importance of making and evaluating multiple/varied prototypes so as to not fixate too early on a narrow set of design features [1]. An example might be .*
- *Iterate means to repeatedly loop through design, prototyping and evaluation so as to incrementally improve one's design and understanding [1]. An example would be a paper prototype of a web site whose evaluation informs the design of an HTML version that is then evaluated.*
- *Principles refers to a number of design principles that span most forms of design [1], such as giving feedback on activated functions and consistency of GUI components [1].*

(b) Prototyping can be split into low fidelity and high fidelity approaches. Explain and compare these approaches. In your answer, give at least one example of each approach.

[6]

Lecture material, but involving simple analysis/comparison
Lo-fi prototypes are generally made using simple materials/techniques such as sketching screen layouts on card or paper [1]. Their main advantage is that are quick to make and easily changed, and cheap to produce. [1]
Hi-fi prototypes are generally made using a programming language such as Macromind Director [1], so the appearance and behaviour of the prototype is much more realistic than in lo-fi [1].
Hi-fi generally takes longer and costs more to make than lo-fi [1]. On the other hand, users may mistakenly think that they have a final product at hand even though performance may be poor/patchy and the graphic design may be rough [1].

(c) Imagine that you are a programmer for a bank. A financial analyst has several different programs for visualising and interacting with a spreadsheet, but would like to combine the techniques of each of his programs so as to make one program with have 'linked views', i.e. so that selection in any one view is reflected in all the others.

Outline a user interface management system (UIMS) that would satisfy this design requirement. In your answer, explain what a UIMS is and also explain the main advantages of your choice.

[9]

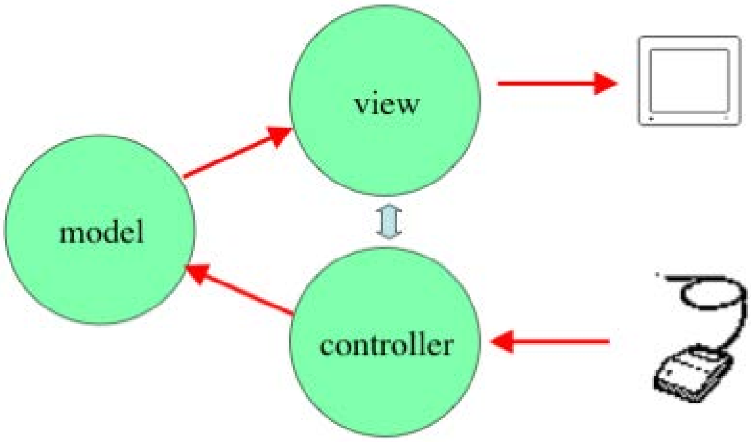
Essay based on unseen design example. Main points:

- *A UIMS is a high-level conceptual architecture for an interactive system [1].*

- *It offers useful data abstraction, in that it separates the internal application logic from the interface representations/interactions [1]. This improves software portability and reuse [1].*

- *The Model-View-Controller UIMS would be suitable here. The model is the application logic, a view renders the model on screen, and a controller processes user input. [2]*

- *Diagram as below [2]*



- *A View and a Controller are usually paired, as input such as a mouse click only makes sense when we know what was clicked on [1].*

- *Having one such View-Controller pair for each of the analyst's techniques would give this linked view effect, as a selection done in any one view would be communicated by the corresponding controller to the shared model, which would then update all the views with the same selection [1].*

2. (a)

(i) What is an entity?

Describe the difference between a strong entity and a weak entity.

[2]

An entity is some aspect of the real world which has an independent existence and can be uniquely identified. A weak entity is an entity whose existence depends on the existence of another entity. A strong entity does not depend on another entity to exist.

[1 mark for definition of entity; 1 mark for distinction between which one requires dependence on existence of other]

(ii) According to the Relational Data Model, explain with examples what is meant by *attributes* and *tuples*.

[4]

[An attribute is a named value type [1]. It represents a column of a relation or table [1]. An example would be the attribute 'surname' in the table 'employee'. [1]

A tuple, or record, is a set of attribute values. [1] Tuples are rows of the relation or table in a database. [1] An example would be (name, dept, ID) in the relation 'employee'. [1]

(up to 2 marks per description)]

(iii) Explain why controlled concurrent access is an important feature of database management systems.

[3]

Controlled concurrent access refers to the process of managing the access of information by potentially many users to the same data at the same time (1). It is important in database management because if many people have access to the same data there has to be a way of keeping track of the correct version of the data, changes to the data, deletion of the data etc (1). Different users may have different privileges for example (1). Different users may have different views on the same data (1). This control prevents disruption to the integrity of the data (1).

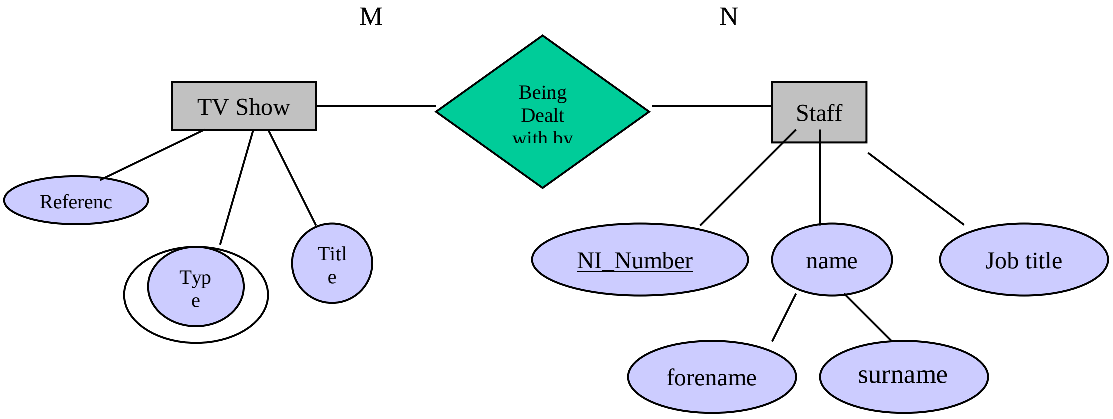
(b) A TV show company keeps data on all of its TV shows. A show has a reference code, a title, and a type (drama, comedy, news). A member of staff has a unique employee code, a name, and a job title. Each show has staff working on it. All staff are assigned to at least one show, but may work on more than one.

Draw an ER diagram that could be used in the development of a database to support the TV show company.

[6]

One mark for the correct entities. One mark for the correct attributes of each entity. One mark for underlining appropriate primary keys. One mark for relationship including 0.5 for correct cardinality. One mark for multivalued attribute Type. One mark for composite attribute Name.

Staff could be totally participative so if they have double lines at this side of the relationship award one mark.



(c)

(i) What is the Power Set of {5,8}?

[1]

(i) $\{\emptyset, \{5\}, \{8\}, \{5,8\}\}$

(ii) For any set, S, is it true that $\emptyset \subseteq S$? Why or why not?

[1]

(ii) True, because ∅ stands for the null set or empty set and the null set is a subset of every set.

(iii) How many elements are in the Cartesian product of A and B, if the cardinality of A is 3 and the cardinality of B is 7?

[1]

(iii) 3 x 7 elements = 21.

(iv) List the elements in the relation R from A = {1,2,7} to B = {2,3,9} where :

$<x,y> \in$ R if and only if $x > y$.

[1]

(iv) {<7,2>, <7,3>}

(d)

(i) What is SQL and what is it used for in databases?

[2]

SQL is a structure Query Language [1]. It is declarative [1] and it specifies what the desired result of a query is [1].

(ii) Assume a relational database for a city restaurant booking facility has two tables as follows:

Customer = {id, name, contact_number}

Booking = {customer, restaurant_number}

Where the customer attribute in Booking is a foreign key referring to the id in Customer.

The following SQL query will return the name of all customers booked into restaurant number “Glasgow53”.

```
SELECT Name
    FROM Customer, Booking
        WHERE (id-customer) AND restaurant-number =  “Glasgow53”)
```

Explain what happens when this query is executed in the database, describing the action of each SQL clause in the order in which it is handled.

[4]

FROM - The cartesian product of the two relations (Customer and Booking) is formed.
WHERE - Rows are eliminated that do not satisfy the where clause (join and direct string comparison).
SELECT - The Select clause is evaluated (ie. the Name attribute is projected).

[3 marks, 1 per stage of execution – additional 1 mark for correct order]

3. **(a)** State two advantages that the digital representations used in modern computers confer over the analog representations previously used.

[2]

Digital representations allow storage, manipulation and transmission of all types of information in a **uniform** manner. Operations such as copying, and viewing the data **do not cause degradation** of information when represented digitally.

(will accept variation on this answer to a maximum of 2 marks)

[2]

(b) What does the abbreviation *ALU* stand for? Explain the purpose of the ALU within the CPU.

[2]

ALU stands for Arithmetic and Logic Unit [1]. The ALU does any calculations required during the execution of a program [1].

(c) Computer architectures have migrated from 32 bits to 64 bits, explain potential reasons for this change.

[2]

Expanding from 32 bits to 64 bits allows for larger address spaces to be accessed giving access to more than 4Gb of memory. [1] 64 bit data paths increase the memory bus bandwidths. [1] Will allow variations on this.

(d) Draw the circuit schematic for an **and** gate based on CMOS transistors and explain the operation of this circuit.

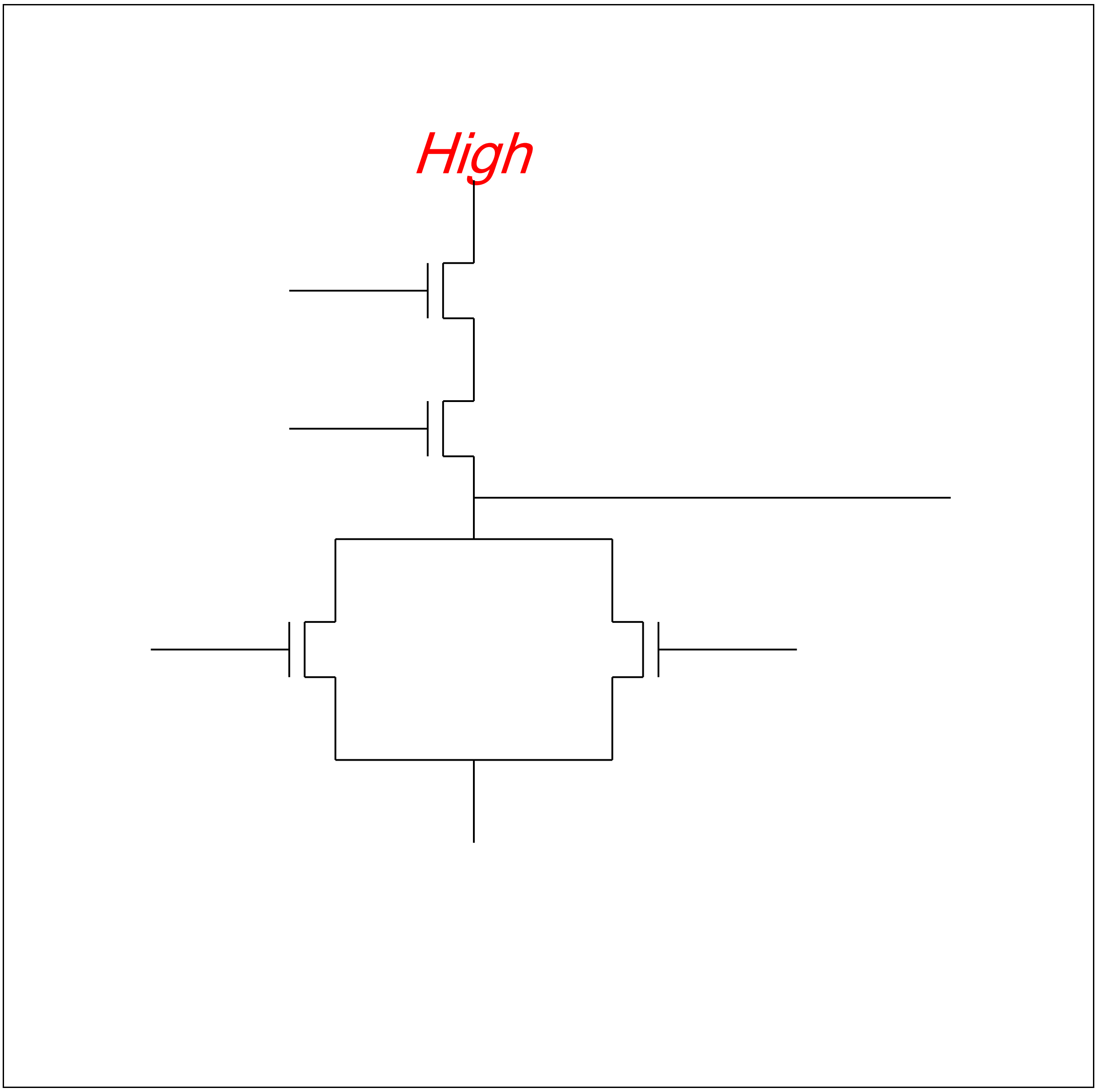
[4]

The circuit consists of two steering networks: one will make a connection, the other will not:

The two upper transistors in series connect *z* to *High* if both *x* and *y* are *High*

The transistors in parallel connect *z* to *Low* if either *x* or *y* is *Low*

Therefore for the and gate to be on both input must be high, and if either input is low the output is low.


High

(e) You are required to design a circuit which, given an input *xyz* representing a 3 bit binary number *n*, produces an output *abc* representing *n*-1. For example, if the input is 110 ($x = 1$, $y = 1$, $z = 0$), representing $n = 6$, then the output is 101, representing 5. If the input is 000 then the output is 111.

(i) Draw a truth table which shows *a*, *b*, *c* as functions of *x*, *y*, *z*.

[3]

x	y	z	a	b	c
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	1
0	1	1	0	1	0
1	0	0	1	1	0
1	0	1	1	1	1
1	1	0	1	0	1
1	1	1	1	0	0

(1 mark for each correct output column.)

(ii) Draw a Karnaugh map for each of *a*, *b*, *c*.

[3]

Karnaugh map for a: (1 mark)

	not(y)	y	y	not(y)
not(x)	1	0	0	0
x	0	1	1	1
	not(z)	not(z)	z	z

Karnaugh map for b: (1 mark)

	not(y)	y	y	not(y)
not(x)	1	0	1	0
x	1	0	1	0
	not(z)	not(z)	z	z

Karnaugh map for c: (1 mark)

	not(y)	y	y	not(y)
not(x)	1	1	0	0
x	1	1	0	0
	not(z)	not(z)	z	z

(iii) Use the Karnaugh maps to work out formulae for *a*, *b* and *c* in terms of *x*, *y* and *z*.

[3]

$$a = \bar{x}\bar{y}\bar{z} + xy + xz$$

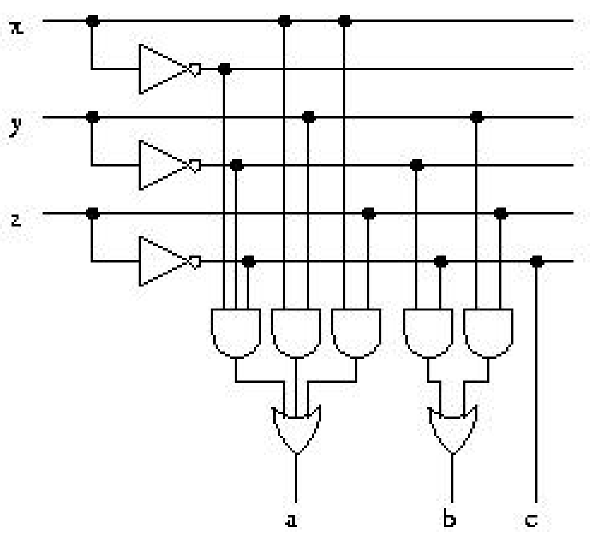
$$b = \bar{y}\bar{z} + yz$$

$$c = \bar{z}$$

(1 mark for each formula.)

(iv) Draw a diagram of the circuit which calculates *a*, *b* and *c* from *x*, *y* and *z*.

[3]



1 mark for each correct part, i.e. calculation of a, b, c.

(v) How could the inputs to this circuit be configured to subtract 2 from a 4 bit binary number?

[3]

If the 4 bit number is *xyzw*, then the result is *abcw* where *a*, *b*, *c* are calculated by this circuit.

4. **(a)** Explain, the difference between multi-tasking and parallel processing and how these concepts relate to each other.

[5]

A multi-tasking operating system gives the appearance of running several processes or tasks simultaneously (1 mark) but in fact time-slices the execution of each task on a single CPU (1 mark) . In parallel processing n (for n >=2) CPUs (or CPU cores) can execute n instruction streams (i.e. processes or tasks) in parallel (1 mark). A parallel processing computer can also multi task each processor (1 mark) to give the appearance of executing a greater number of processes than physically present processors. (1marks)

(b) Explain the concept of *interrupts* and how these are used within operating systems, for example to implement multi-tasking.

[4]

An interrupt is a hardware generated signal that causes the CPU to halt execution of a current program[1] and save its current state [1] and change its program counter to enter the operating system.[1] Inside the operating system, the task scheduler decides what to do next depending upon the type of interrupt,[1] for example it may cause execution to be transferred to another program to afford a "time-slice" of execution. [1] Interrupts are generated by the hardware on a regular basis [1] or by hardware events that must be serviced such as mouse clicks or by communications buffers filling or emptying.[1] (I will accept variations to 4 marks)

(c) Explain why there isn't a central database which stores the IP address for each domain name in the internet, and briefly describe how this information is actually stored and accessed.

[6]

A central database would generate too much traffic (1 mark) and failure would be disastrous (1 mark) and it would be enormous, so performance would be a problem (1 mark). Actually have a hierarchy of domain name servers, each of which is an authority for a domain (1 mark), and knows which server is an authority for the next domain up (1 mark). DNS lookup requests return either an IP address or the address of another server to try (1 mark).

(d) Do you think that a consistent user interface based on a web browser would be useful for all internet applications? Justify your answer, using email as a specific example.

[10]

Marks for identifying some or all of the following issues, and perhaps others, and for general style and coherence. This is an open-ended question and answers which bring in other topics from the course should be rewarded.

Consistency: sounds good but can it be achieved? - there is huge variation in style of interface among web sites. Performance: if repeated interaction with a server is necessary in order to e.g. read a sequence of emails, this might be much slower than downloading all email and using a local email application; similarly for composing/sending. Security: web applications requiring security tend to use more complex interaction - entering/leaving secure pages; a dedicated application could hide more of this. Ease of access/installation: web applications can be used from any browser device including public telephones etc.; there's no need to install or upgrade special software. Applets: in principle perhaps any application could be converted into an applet and used via a web browser, but there are other issues - download time for the applet, technical restrictions on the communication capabilities of applets; in any case this approach wouldn't lead to uniformity of interfaces. Email as an example: making points based on their own use of a web email system during labs.