

Appendix 1: Test 1, Stage 1. Individual metaphors.

Please write down the meaning(s) of the words below. Try to use only English to explain first, if you find that you really cannot express yourself well, you are allowed to use Chinese or both languages together. While you are writing down the meanings of the words, please say aloud whatever comes into your mind. Many thanks.

1. **candidate**

2. **subordinate**

3. **bug**

4. **tree**

Appendix 2: Test 1, Stage 2. Words in context.

Please write down the meaning(s) of the words underlined (longer texts will be provided upon request). Try to use only English to explain first. If you find that you really cannot express yourself well, you are allowed to use Chinese or both languages together. While you are writing down the meanings of the words, please say aloud whatever comes into your mind. Many thanks.

Example 1: Anthropomorphic metaphor: candidate

Note the use of a composite key in Figure 8.2. Composite keys are candidate keys that contain more than one attribute. They are needed when there is no single attribute candidate key.

Example 2: Anthropomorphic metaphor: subordinate

Inversion of authority: a special case of control coupling (q.v.) in which a subordinate module communicates information to a superordinate module for the explicit purpose of influencing the execution of that super-ordinate.

Example 3: Organic metaphor: bug

Because they were not formally designed, they cannot be precisely repeated, and no one is sure whether there was a bug or not. After the bug has been ostensibly corrected, no one is sure that the retest was identical to the test that found the bug.

Example 4: Organic metaphor: tree

To implement the doctor-imitating program, we begin with DOCTOR, the procedure that couples you to the Doctor transition tree: your job is to implement DOCTOR-TREE, a transition tree procedure, based on the sample dialog, that recognises sentences and returns appropriate responses.

Appendix 3: Test 1. Supplementary information.

Please write down the meaning(s) of the words below. Try to use only English to explain first. If you find that you really cannot express yourself well, you are allowed to use Chinese or both languages together. While you are writing down the meanings of the words, please say aloud whatever comes into your mind. Many thanks.

Word 1: *candidate*

This side effect is an insertion anomaly. Consider, as a second example, the records of Figure 8.2. They contain data about book dealers, titles, publishers, and costs. Given file STOCK, one could not answer queries about the costs of a particular book title unless a dealer had that title in stock. (Note the use of a composite key in Figure 8.2. Composite keys are *candidate* keys that contain more than one attribute. They are needed when there is no single attribute candidate key.) Modification, deletion, and insertion anomalies can be minimised by design. Try the update modification we attempted on STUDENTS in Figure 8.1 on the two record specifications of Figure 8.3. Note that the probability of a modification, deletion, or insertion anomaly has been reduced. This is because the specifications in Figure 8.3 are less complex: they represent single associations, not multiple associations.

MicroConcord search SW:	<i>candidate</i>
Sort:	1R/SW unshifted, 999 characters per entry (with modifications).
Data from the following files:	DB038-2.ASC

Word 2: *subordinate*

Informational cluster: a set of modules that have exclusive right of access to a particular item or items of data (normally used when the data has a complex structure or has sensitive security).

Intermediate file: a temporary file used to communicate data between job steps or, sometimes, jobs. Inversion of authority: a special case of control coupling (q.v.) in which a *subordinate* module communicates information to a superordinate module for the explicit purpose of influencing the execution of that superordinate.

Job: a sequence of one or more job steps that is activated by a human operator (or another agent outside of the executing computer); the smallest unit that can be activated in this way (see also job step).

Job step: one main program that has (optionally) a hierarchy of one or more subprograms and that is activated by an operating system; the smallest unit that can be activated in this way.

MicroConcord search SW:	<i>subordinate</i>
Sort:	1R/SW unshifted, 999 characters per entry (with modifications).
Data from the following files:	SE033-3.ASC

Word 3: *bug*

Although programmers, testers, and programming managers know that code must be designed and tested, many appear to be unaware that tests themselves must be designed and tested—designed by a process no less rigorous and no less controlled than that used for code. Too often, test cases are attempted without prior analysis of the program's requirements or structure. Such test design, if you can call it that, is just a haphazard series of ad-lib cases that are not documented either before or after the tests are executed. Because they were not formally designed, they cannot be precisely repeated, and no one is sure whether there was a **bug** or not. After the bug has been ostensibly corrected, no one is sure that the retest was identical to the test that found the bug. Ad-lib tests are useful during debugging, where their primary purpose is to help locate the bug, but ad-lib tests done in support of debugging, no matter how exhausting, are not substitutes for designed tests. The test-design phase of programming should be explicitly identified. Instead of "design, code, desk check, test, and debug", the programming process should be described as: "design, test design, code, test code, program inspection, test inspection, test debugging, test execution, program debugging, testing."

MicroConcord search SW:	<i>bug</i>
Sort:	1R/SW unshifted; 999 characters per entry (with modifications).
Data from the following files:	SE013-1.ASC

Word 4: *tree*

Problem 28-4: It is easy to write a simple version of a program that seems to interact with people at a terminal much like the way one type of psychiatrist interacts with people on the couch. To implement the doctor-imitating program, we begin with DOCTOR, the procedure that couples you to the Doctor transition *tree*. Your job is to implement DOCTOR-TREE, a transition tree procedure, based on the sample dialog, that recognises sentences and returns appropriate responses. Here is a fragment you can start with. Your procedure will be similar in spirit to the key procedure in DOCTOR, also known as ELIZA, a classic program that seemed to understand sentences.

MicroConcord search SW:	<i>tree</i>
Sort:	1R/SW unshifted; 999 characters per entry (with modifications).
Data from the following files:	PR048-3.ASC

Appendix 4: Test 3, Stage 2. Allusions and facetious words.

Please write down the meaning(s) of the words below. Try to use only English to explain first. If you find that you really cannot express yourself well, you are allowed to use Chinese or both languages together. While you are writing down the meanings of the words, please say aloud whatever comes into your mind. Many thanks.

Sentence 1: When a data record is recorded, it is often preceded by a preamble and followed by a *postamble*.

Sentence 2: Creating task overlaps and *underlaps* through poor organisation, delegation, and task monitoring ...

Sentence 3: You'll have to play *catch-up*, and in the systems game, that may be difficult.

Sentence 4: The *turbulence* in the British telecommunications airspace is caused by the political desire to deregulate a monopolistic industry.

Sentence 5: This question is an example of what I call the "*Las Vegas*" mentality ... When you carry out a transform analysis, you don't have to stake everything on a single throw.

Sentence 6: Another possibility is to play *Robin Hood*, taking space from the tables with much room and giving it to the tables with little room.

Note: If you need longer texts to understand the words in italics, please read Appendix 5.

Appendix 5: Test 3. Supplementary information.

The longer texts below are intended to give you more information to understand **sentences 1–6**. Please read them carefully and try to explain the meaning of the words in italics. Many thanks.

Sentence 1: *postamble*

This theoretical capacity depends on the distance between tracks and the distance between bits on the track, that is, on the bits per square inch. Figure 5.44 summarises the trend over time. This number corresponds to the absolute maximum storage possible, a capacity that is never realised in practice. When a data record is recorded, it often is preceded by a preamble and followed by a *postamble*. The preamble initialises the read electronics; the postamble discharges and turns off the write electronics. The postamble may also contain parity and cyclic redundancy error-code check bits. The preamble and postamble both require space and thus reduce the storage capacity of the device. It should also be noted that many direct-access devices are able to address a specific sector of a specific track.

MicroConcord search SW:	<i>preamble</i>
Sort:	1R/SW unshifted, 999 characters per entry (with modifications).
Data from the following files:	AR010-2.ASC

Sentence 2: *underlaps*

MANAGEMENT QUALITY Poor management can increase software costs more rapidly than any other factor. Each of the following mismanagement actions has often been responsible for doubling software development costs: Assigning the wrong (combination of) people to project jobs; Creating task overlaps and *underlaps* through poor organisation, delegation, and task monitoring; Demotivating people by unnecessarily poor working conditions and failure to reward good performance; Bringing large numbers of people onto the project before there is a clear understanding of their responsibilities; Failing to prepare needed resources: computer time, terminals, communications, test data, support software; Failing to validate software requirements and design specifications, and to identify and resolve high-risk elements early.

MicroConcord search SW:	<i>overlaps</i>
Sort:	1R/SW unshifted, 999 characters per entry (with modifications).
Data from the following files:	SE014-3.ASC

Sentence 3: *catch-up*

Old software can give your competition an edge. For example, what happens when TWA can confirm your return seat and issue a boarding card? You don't have to stand in line for your return flight. For the same price they provide a convenience not provided elsewhere. What if your competition has a better inventory system that keeps inventory levels down and can immediately find an item in stock without the usual clerical search? You'll have to play ***catch-up***, and in the systems game, that may be difficult. Old software and technology also generate personnel problems. Data processing people tend to follow technology, so your old systems won't hold them on the job unless they like the security. Delaying investments in new software and hardware can affect the corporation competitively and financially. Current management bonus programs tend to make managers focus on short-term solutions that adversely affect the long-term survival of the company.

MicroConcord search SW:	<i>catch-up</i>
Sort:	1R/SW unshifted, 999 characters per entry (with modifications).
Data from the following files:	SEO26-2.ASC

Sentence 4: *turbulence*

Value-added network services (VANS), defined as "applicable systems," have also been liberalised, with licences being awarded by the Government but not by BT. Two competing cellular radio operators have been licensed, of which BT is only a part shareholder in one. Finally, the Government is examining the possibility of allowing resale of BT private circuits - the final liberalisation. The ***turbulence*** in the British telecommunications airspace is caused by the political desire to deregulate a monopolistic industry. But the turbulence is constantly increased by the difficulties of defining and managing the telecommuni-cations business. Although it is technically

possible to separate out the conduit-content components of the telecommunication business-network (transmission and switching), enhanced or value-added services (data processing, protocol conversion, customer equipment - there are stronger forces at work which are driving the components together and blurring boundaries between content and conduit. It is this which continues to generate turbulence.

MicroConcord search SW:	<i>turbulence</i>
Sort:	1R/SW unshifted, 999 characters per entry (with modifications).
Data from the following files:	IS004-1.ASC

Sentence 5: *Las Vegas*

Q: What if I hire a new boss and then discover that the job he has to do is utterly trivial? or, what if I decide on the promote-a-boss approach and I pick the wrong bubble?

A: This question is an example of what I call the "**Las Vegas**" mentality exhibited by some designers. When you carry out transform analysis, you don't have to stake everything on a single throw. You should try out two or three different bosses. It takes only a few minutes to sketch the top of the structure chart. By the time you have completed the sketch, you can tell which approach will give you a design that is most easily improved by means of the Structured Design criteria.

MicroConcord search SW:	<i>Las Vegas</i>
Sort:	1R/SW unshifted, 999 characters per entry (with modifications).
Data from the following files:	SE033-2.ASC

Sentence 6: *Robin Hood*

Consider what happens if a program has an exceptionally large number of variables. The chunk of address space allocated for the symbol table may fill up, but there may be lots of room in the other tables. The compiler could, of course, simply issue a message saying that the compilation cannot continue due to too many variables, but

doing so does not seem very sporting when unused space is left in the other tables. Another possibility is to play **Robin Hood**, taking space from the tables with much room and giving it to the tables with little room. This shuffling can be done, but it is analogous to managing one's own overlays - a nuisance at best and a great deal tedious, unrewarding work at worst. What is really needed is a way of freeing the programmer from having to manage the expanding and contracting tables, in the same way that virtual memory eliminates the worry of organizing the program into overlays.

MicroConcord search SW:	<i>Robin Hood</i>
Sort:	1R/SW unshifted, 999 characters per entry (with modifications).
Data from the following files:	DB103-2.ASC