



Do not award half marks.

In all cases give credit for appropriate alternative answers.

Question 1 (Compulsory)

- (a) Identify and describe *two* skills required of all software developers. [4]

Software skills (1 mark): this is the nuts and bolts of actually writing the software, which typically includes software construction, programming, designing the architecture of the software, and so on (1 mark). Project management skill (1 mark): this refers to the how-to control the development of software, e.g., budgetary constraints and human resources, and subsequent maintenance of software.

- (b) For each of the four statements given below, identify the design activity being referring to.
- (i) The design activity that involves the modular arrangement of parts and structure in the software program. [1]
 - (ii) The design activity that concerns the interaction mechanisms of program systems within the software. [1]
 - (iii) The design activity that concerns the creation of valid and usable data structures within the software. [1]
 - (iv) The design activity that is concerned with the series of steps taken to accomplish some function in the program. [1]
- (i) **Architectural (1 mark).**
 - (ii) **Interface (1 mark).**
 - (iii) **Data (1 mark).**
 - (iv) **Procedural (1 mark).**

- (c) Fourth generation techniques are one example of a software development methodology. Some of the commonly mentioned characteristics of fourth generation techniques include "they are easy to use" and "they enable users with little software development skills to create programs". Based on your understanding of this class of technique, comment on how these characteristics are true. [3]

Fourth generation techniques essentially combine procedural and non-procedural characteristics, and allow for programs to be written by end users without requiring them to possess detailed knowledge of programming. This is because the methods of creating programs using 4th generation techniques are typically intuitive, and natural for the human being, e.g. through graphical user interfaces. In addition, they typically support the automatic creation of source code based on some input and output specification.

0 marks should be awarded for an obviously incorrect or non-informative explanation.

1 mark should be awarded for an answer that although has one good point, does not otherwise indicate any knowledge of the advantages of fourth generation techniques.

2 marks should be awarded for an explanation that has more than one good point, but still can be improved upon.

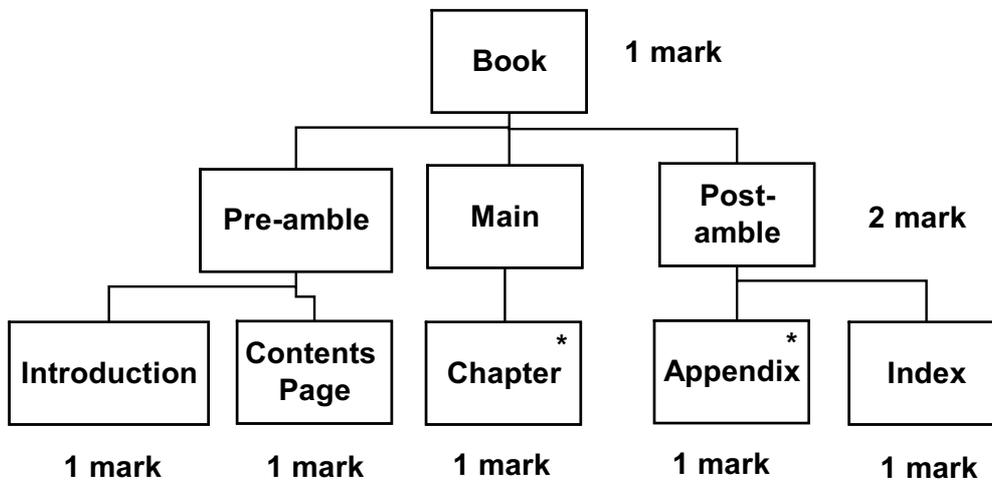
3 marks should be awarded for a clearly excellent answer that demonstrates good competence and understanding.

- (d) The following text describes in some detail an object.

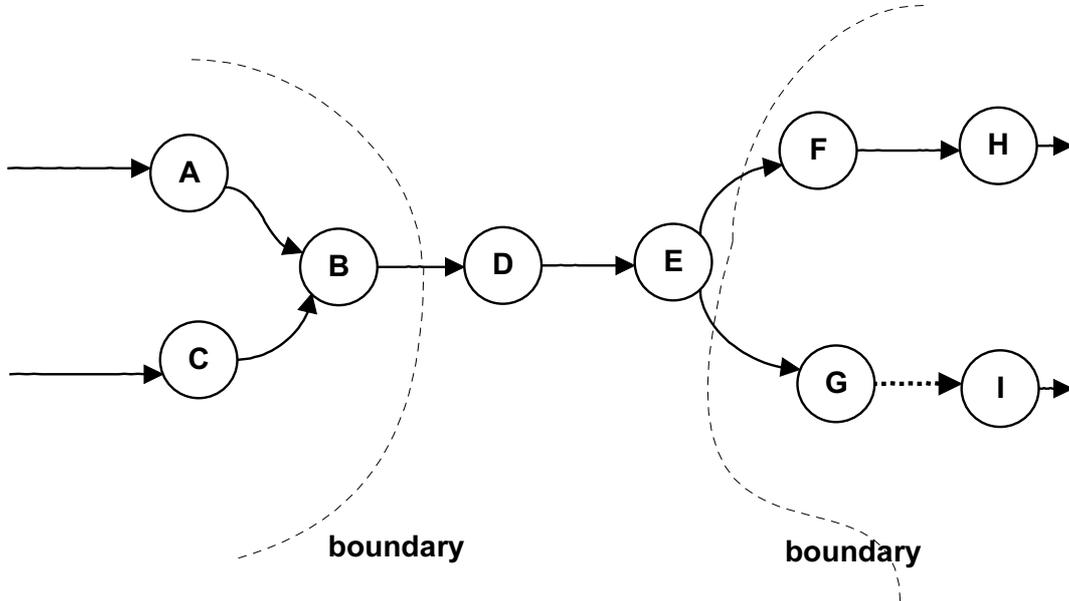
The object concerned is a book. The book comprises of three main sections: the pre-amble section, the main section, and lastly the post-amble section. The pre-amble section comprises of both the introduction page, followed by a contents page. The main section comprises of many chapters. In the post-amble section, several appendices will be found, and also one index.

Making use of Jackson's notation, draw a simple structure to represent this object.

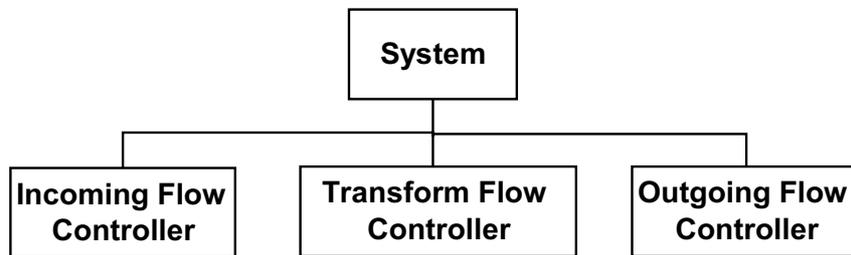
[8]



- (e) The following DFD fragment is said to exhibit transform flow characteristics, with boundaries indicated.

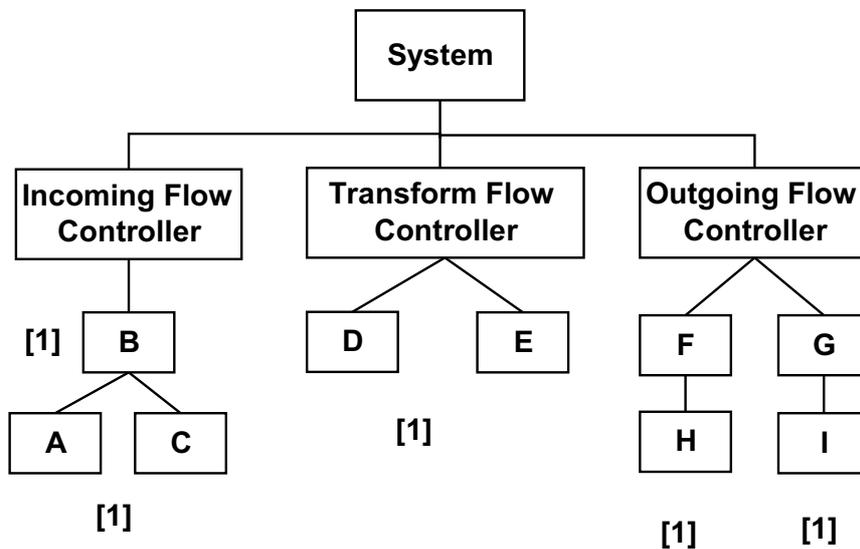


Show how you would complete the following program structure so that the DFD given above maps to it.



[5]

Answer



- (f) The Formal Technical Review (FTR) is carried out during development with the specific intent of discovering errors in development so far. The three possible outcomes of the FTR are as follows: the product is accepted and development proceeds; the product is rejected and development stops; the product is accepted conditionally and development proceeds. Identify the conditions that would result in each of these outcomes. [3]

Accept the product: when no errors were identified (1 mark). Reject the product: when major errors have been identified (1 mark). Accept the product provisionally; when minor errors have been identified (1 mark).

- (g) A program under testing suspends execution suddenly. The following three scenarios refer to different debugging approaches. For each scenario, identify the debugging approach that it is closest to.

Scenario 1: The section of code is roughly sequential from first to last line. Since the program suddenly suspends execution roughly midway in program execution, the programmer suspects that the error must have occurred around there. Hence, from the point of suspension, code is checked, and gradually trace backwards according to program execution until the error is found.

Scenario 2: The programmer checks every line of code in the program, from first to last line, looking for the error causing the program suspension.

Scenario 3: The programmer is familiar with the code, and based on evidence gathered upon program suspension, suspects that the error is in line 10, 35 or 120 of the program. He investigates specifically and only each of these lines of code, eventually discovering the error at line 120. [3]

Scenario 1: back-tracking (1 mark).

Scenario 2: brute force (1 mark).

Scenario 3: cause elimination (1 mark).

Question 2

- (a) A director of a company has decided to computerise his daily transactions, currently carried out in a manual mode of operation. He has already defined a complete set of general objectives on how his software should operate, but he has no ideas on the detailed input, processing or output requirement for the intended systems. He then approaches a newly employed software analyst (SA), instructing him about his proposed system. Since this is a new system, the SA isn't exactly sure about the efficiency of the algorithms involved either, much less the required operating system to be used and the types of interfaces for input and output. Besides, the customer has also insisted on seeing the interface designs urgently so that any differences, modification or disagreement can be rectified quickly before letting the project go ahead. The SA is expert in program testing and has promised the customer that the product will undergo systematic testing before delivery.

Which paradigm would you advise the SA to use? Justify your answer. [3]

Combining paradigms (1 mark). The customer is unsure of the input, processing and output requirements (1 mark). The SA does not know the efficiency of the algorithm, OS to be used or interface designs (1 mark).

- (b) You are the manager of a local software development company. Two scenarios (given below) have been put to you.

Scenario 1: You are to develop a simple information system that will keep track of inventory and purchases of stationary in a bookshop.

Scenario 2: You need to write a device driver for a modem; this device driver will communicate between the host operating system and the modem.

For each of the two scenarios above, identify what class of programming language you would use for the scenario, giving, for each scenario, *two* reasons why you chose that class of language. [6]

The first scenario requires the use of a fourth generation language [1], since the application in question is an information system/database [1]. The application is also query-driven [1], since the addition and deletion of records is required in the application.

The second scenario requires the use of a second generation, low level language [1]; low level programs need to be hand coded line by line for optimum effect [1], and also because device drivers communicate directly [1] with the native instruction set of the hardware device.

- (c) Data type checking is concerned with the manner in which different data types can be manipulated in the same statement. Consider the simplified scenario:

Three programming languages, "BASIC-2", "PASCAL-2", and "FOXPRO-2", make use of the following data type checking:

- "BASIC-2": typeless- the programming language has no explicit means for data typing and does not enforce type checking.
- "PASCAL-2": mixed-mode- the programming language allows for mixing of different data types, and then converts operands of different but compatible types.
- "FOXPRO-2": strong-type checking- the programming language only allows operations on data objects to be performed if they are of the same data type.

A program makes use of the following variables:

integers a;
floating point numbers b;
characters c;

- (i) Consider the operation

a+b;

For each of the three programming languages given above, describe whether the operation will be allowed or disallowed, describing briefly any conversion of operands that may occur.

[3]

- (ii) Consider the operation

a+c;

For each of the three programming languages given above, describe whether the operation will be allowed or disallowed, describing briefly any conversion of operands that may occur.

[3]

- (i) **The operation is allowed in BASIC-2 (1 mark); the operation is allowed in PASCAL-2 (a is converted to a floating point number) (1 mark); the operation is disallowed in FOXPRO-2 (1 mark).**
- (ii) **The operation is allowed in BASIC-2 (1 mark); the operation is disallowed in PASCAL-2 (1 mark); the operation is disallowed in FOXPRO-2 (1 mark).**

Question 3

- (a) The specification of a project frequently forms the basis for correct software development to follow. Two of the benefits of the specification document are frequently stated as follows:
- It can act as a formal contract for software development between users and developers.
 - It can be a means of assessing quality of software.

Based on your understanding of the specification document, describe in some detail how each of these can be true. [2]

Essentially, as software development is a business transaction between the user and developer, it would make sense for the terms of this transaction to be clearly and formally specified in black and white, i.e. the expectations of the program to be developed.

At the end of development, this specification will be in turned used to ascertain the quality of the product developed, since for all purposes quality can be defined as a product having met this pre-defined expectations.

**0 marks for an obviously incorrect or non-informative explanation.
1 mark for an answer that shows some understanding.
2 marks for an explanation that clearly shows maturity and an awareness of the characteristics of the specification.**

- (b) A number of problems are often encountered in the activities associated with requirements analysis. Describe *three* of these problems. [3]

One mark should be awarded for each such problem described (up to a maximum of three marks). Examples include the following.

- **Noise (e.g. miscommunication) and omission of information invariably occurs during communication.**
- **Different levels of communication exist between an analyst and customer**
- **Successful acquisition of information cannot be guaranteed, given that analysts have difficulties getting pertinent information.**
- **Handling large and complex problems.**
- **Accommodating changes that occur during and after analysis.**

- (c) The Facilitated Application Specification Technique (FAST) is a communication technique that takes place between the analyst and customer. Write down any *three* basic guidelines associated with this technique. [3]

One mark should be awarded for each such problem described (up to a maximum of three marks). Examples include the following.

- **Meeting is conducted at a neutral site.**
- **Rules for preparation and participation are established.**
- **An agenda is suggested to cover all the important points. The agenda must be formal enough to cover all important points, but informal enough to encourage the free flow of ideas.**
- **A "facilitator" is appointed to control the meeting. A facilitator is the controller, overall chairman of the meeting.**

(d) Consider the following scenario.

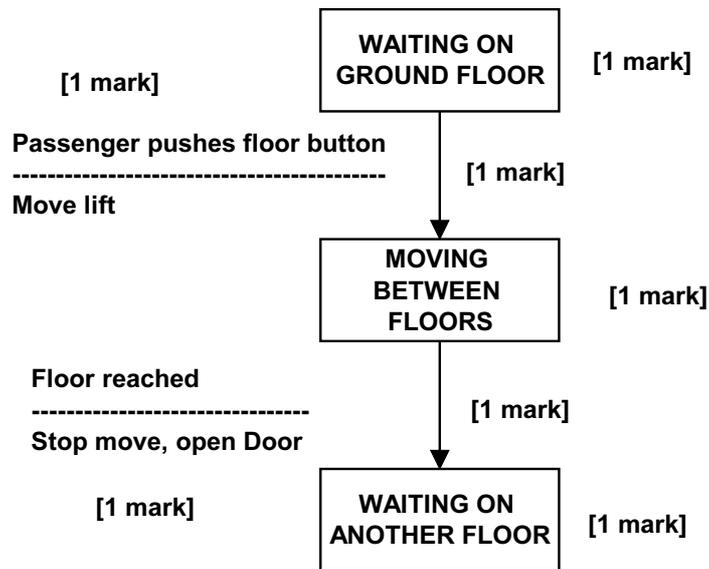
A lift system operates in possible states: “WAITING ON GROUND FLOOR”, “MOVING BETWEEN FLOORS”, and “WAITING ON ANOTHER FLOOR”. The following events and subsequent actions can take place from state to state:

When the lift is in the state “WAITING ON GROUND FLOOR”, and the event “PASSENGER PUSHES FLOOR BUTTON” takes place, the lift will carry out the action “MOVE LIFT”, and the state will change to “MOVING BETWEEN FLOORS”.

When the lift is in the state “MOVING BETWEEN FLOORS”, and the event “FLOOR REACHED” takes place, the lift will carry out the action “STOP MOVE, OPEN DOOR”, and the state will change to “WAITING ON ANOTHER FLOOR”.

Construct a simple State Transition Diagram to represent the above scenario. You should not need to assume any other information besides that provided above.

[7]



Question 4

- (a) The software developer and an independent test group will participate in software testing. Based on your own understanding, explain why the participation of both parties is important. [2]

A broad descriptive answer can be found in the study material:

The software developer is always responsible for testing the individual units (modules) of the program, ensuring that each performs the function for which it was designed. In many cases, the developer also conducts integration testing - the testing step that leads to the construction (and test) of the complete program structure. Only after the software architecture is complete does an independent test group become involved.

The role of an independent test group (ITG) is to remove the inherent problems associated with letting the building test the thing that has been built. In other words, they will remove the conflict of interest that will otherwise be present. After all, the personnel in the independent group are paid to find errors.

0 marks should be awarded for an answer that demonstrates no understanding.

1 mark should be awarded for an answer that demonstrates some understanding, but clearly needs improvement.

2 marks should be awarded for an answer that clearly demonstrates of good understanding.

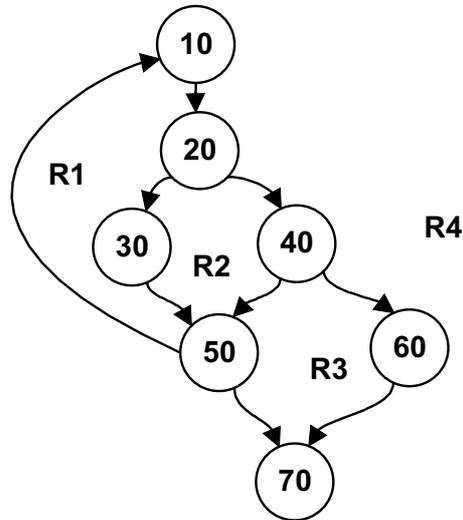
- (b) Consider the following section of code:

```
10    Print "Hello"
20    If Condition C1 is true, then GOTO 30; Else GOTO 40
30    Accept Input; then GOTO 50
40    If Condition C2 is true, then GOTO 50; Else GOTO 60
50    While Condition C3 is true, GOTO 10; Else GOTO 70
60    Accept Input; then GOTO 70
70    Print "End of program"
```

Draw a labelled flow graph for the above source code.

Based on the Basis Path Testing technique, calculate the value of cyclomatic complexity using all three methods of calculation. Ensure that all working is clearly shown for the three methods (e.g. predicate nodes and regions clearly highlighted).

Based on the value of cyclomatic complexity calculated, derive the set of basis test paths. [8]



1 mark for the diagram

$V(G) = \text{no. of regions} = R1 \text{ to } R4 = 4$ [1 mark]

$V(G) = \text{no. of Predicate nodes} + 1$
 $= P(20, 40, 50) + 1 = 3 + 1 = 4$ [1 mark]

$V(G) = E - N + 2 = 9 - 7 + 2 = 4$ [1 mark]

Therefore 4 paths.

Basis Set:

- 1) 10-20-30-50-70 (1 mark)
- 2) 10-20-30-50-10-20-30-50-70 (1 mark)
- 3) 10-20-40-50-70 (1 mark)
- 4) 10-20-40-60-70 (1 mark)

Be careful not to penalise for consequential error in this question.

- (c) Distinguish between the terms *symptom* and *cause* in relation to the debugging activity.

[2]

Symptom normally refers to the visible effects of the error (1 mark), whereas cause refers to the actual line of erroneous code (1 mark).

- (d) *Debugging* and *testing* are different activities, which are - nevertheless - related. Distinguish between these activities, and also comment on any possible relationship between the two. [3]

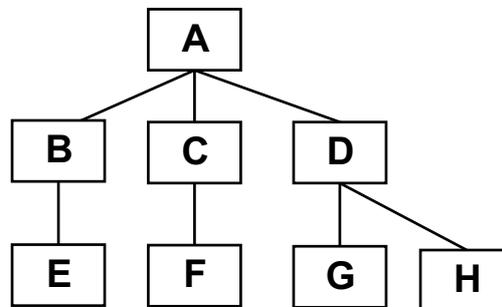
Testing is concerned with the identification of errors [1 mark], whereas debugging with the correction of code that cause these errors [1 mark].

Debugging can only take place in the event of successful testing has taken place, since without the discovery of new errors, debugging cannot take place (1 mark).

[Alternative answers, particularly for the second part of the answer, should be awarded due credit.]

Question 5

- (a) Consider the following structure.



Assume that module D comprises 100 programming language statements. On the basis of this assumption, calculate the value of Henry-Kafura Metric (HKM) for module D. Identify specifically any fan-in or fan-out values you are making use of as well. Make sure that all working is suitably shown. [4]

Fan-in for module D = 1 (1 mark).

Fan out for module D = 2 (1 mark).

HKM = length (i) X [fan-in + fan-out] squared (1 mark)
= 100 * [1 + 2] squared
= 100* 9
= 900 [1]

- (b) For each of the following types of measurement, state whether it is a direct measurement or an indirect measurement.

(i) Lines of code in a program.

(ii) Amount of random access memory in bytes required for software execution.

(iii) Number of discovered errors per line of code. [3]

(i) is a direct measure (1 mark)

(ii) is a direct measure (1 mark)

(iii) is a direct measure (1 mark)

- (c) Web sites can be simple or complex, and be document or application-centric. Identify the four types of web sites in terms of these aspects, and describe them in relation to its complexity and whether the site is document or application-centric.

[8]

- **Static Web Sites (1 mark):** in its most basic form, this web site is little more than a collection of static documents created in HTML and tied together with links (1 mark).
- **Static with Form-Based Interactivity (1 mark):** Forms are generally used on these sites to collect information from the user, including comments or requests for information (1 mark).
- **Sites with Dynamic Data Access (1 mark):** it is possible for a Web site to provide a front end for accessing a database; via a Web page, users can search a catalogue or perform queries on the contents of a database (1 mark).
- **Dynamically Generated Sites (1 mark):** Marketers often want to provide customized pages and content-based on user preferences in order to foster a one-to-one marketing relationship; these web sites dynamically create documents with content often generated from a database (1 mark).

- END OF PAPER -