Do not award half marks.
In all cases give credit for appropriate alternative answers.

Question 1 (Compulsory) (30 marks)

(a) The following are three basic software development methodologies. List **one** advantage and **one** disadvantage for each methodology.

(i) Classic model.

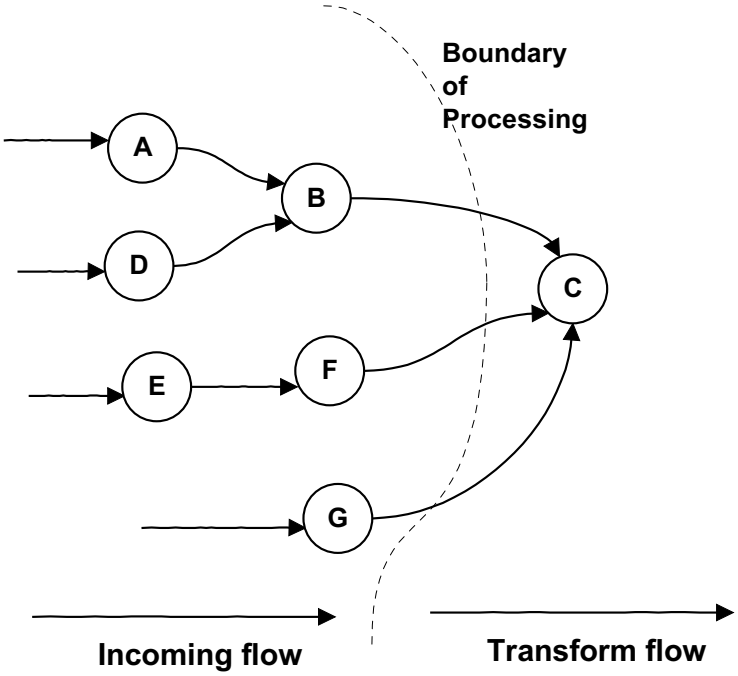
(ii) Prototyping.

(iii) 4th generation techniques. [6]

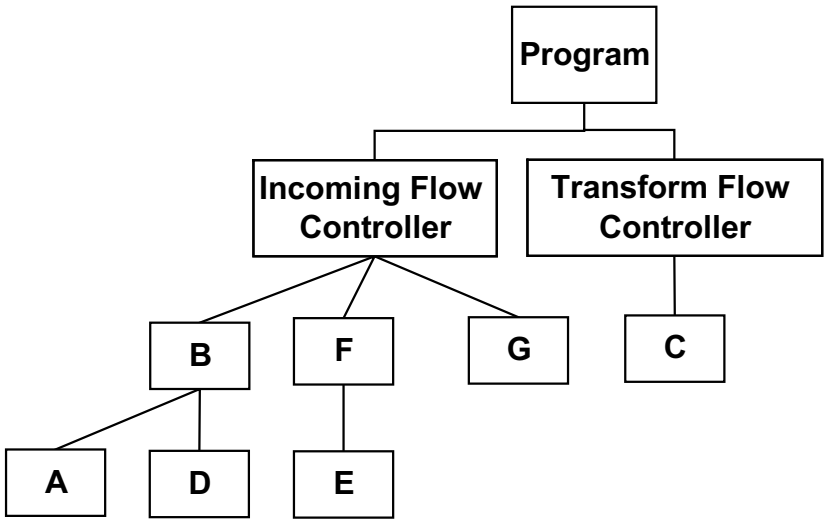
	Advantages	Disadvantage
Classical Life-Cycle	**Sequential method of development, well-established, suits wide range of applications well.**	**Users change requirements, and no contingency for this.**
Prototyping	**The concept of a visual "model" that allows users and developers to refine requirements.**	**Iteration of prototype from continually changing requirements will result in wasted time and effort.**
4th Generation Techniques	**Broad range of tools allows for easy creation and modification of software.**	**Limited application domain.**

(1 mark for each characteristic. Alternative answers are quite possible here)

(b) The transform analysis design technique takes a data-flow diagram, and maps it into a simple program structure with the aid of a pre-defined template. Consider the following data-flow diagram showing *incoming flow*.



Map all the processes *A to G* into a program structure. [8]



One mark should be awarded for the correct placement of each process, and one mark should be awarded for representing the relationship between the processes and the flow controllers.

(c) The following are characteristics of programming languages. Describe each of them.

(i) Uniformity.
(ii) Compactness.
(iii) Ambiguity.
(iv) Tradition.
(v) Source code portability.
(vi) Availability of development tools. [6]

(i) Uniformity indicates the degree to which a language uses consistent notation, applies arbitrary restrictions (1 mark).

(ii) Ambiguity refers to the situation in which a programming language is perceived by the programmer in one way, but the compiler always interprets the language in another way (1 mark).

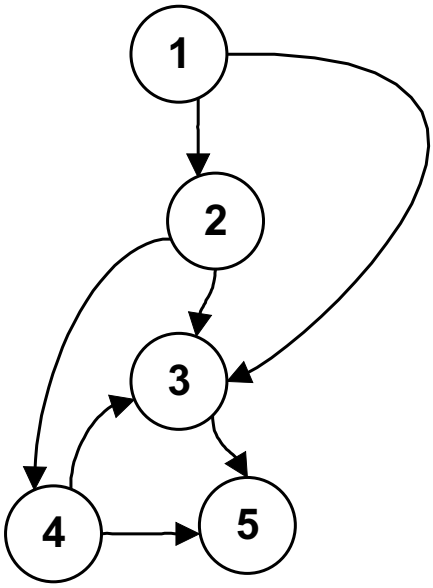
(iii) Compactness indicates the amount of code-oriented information that must be recalled from human memory (1 mark).

(iv) Tradition refers to the fact that a programmer with experience in one form of language will find it easy to pick up another language that has the same sort of constructs in the former (1 mark).

(v) Source code portability generally refers to whether source code may be transported from processor to processor and compiler to compiler with little or no modification (1 mark).

(vi) Availability of development tools can shorten the time required to generate source code and can improve the quality of code (1 mark).

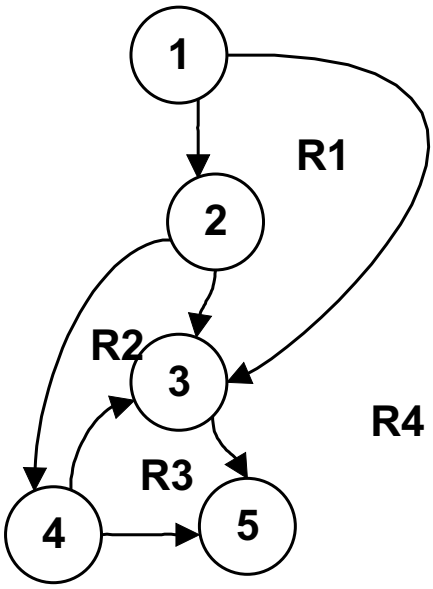
(d) Consider the following flow graph.



(i) Copy the above diagram into your answer sheet, and based on this flow-graph, derive the value of V(G). Show all three methods of calculation, with full working. [6]

(ii) Based on this value of V(G), derive the basis set of test paths. [4]

Regions method: V(G) = no. of distinct regions = R1 to R4 = 4



Regions must be clearly identified before the mark can be awarded.
V(G) = predicate nodes + 1 = Predicate nodes (1, 2, 4) + 1 = 3 + 1 = 4
V(G) = edges – nodes + 2 = 7 edges – 5 nodes + 2 = 4
(2 marks should be awarded for the correct application of each method.)

Basis set:
Path 1: 1-3-5
Path 2: 1-2-3-5
Path 3: 1-2-4-5
Path 4: 1-2-4-3-5

Alternative basis sets are possible; as long as each path in the set of four is examining a previously untested edge.

4 marks for complete correctness of paths
3 mark if one mistake is made in path
2 mark if two mistakes are made in paths
1 mark if three mistakes are made in paths

Question 2

(a) Define what is meant by the term *Software Engineering*. [2]

Software Engineering is the establishment and use of sound engineering principles (1 mark) in order to obtain, in an economical fashion, software that is reliable and works efficiently on real machines (1 mark).

(b) List *three* problems associated with the software crisis. [3]

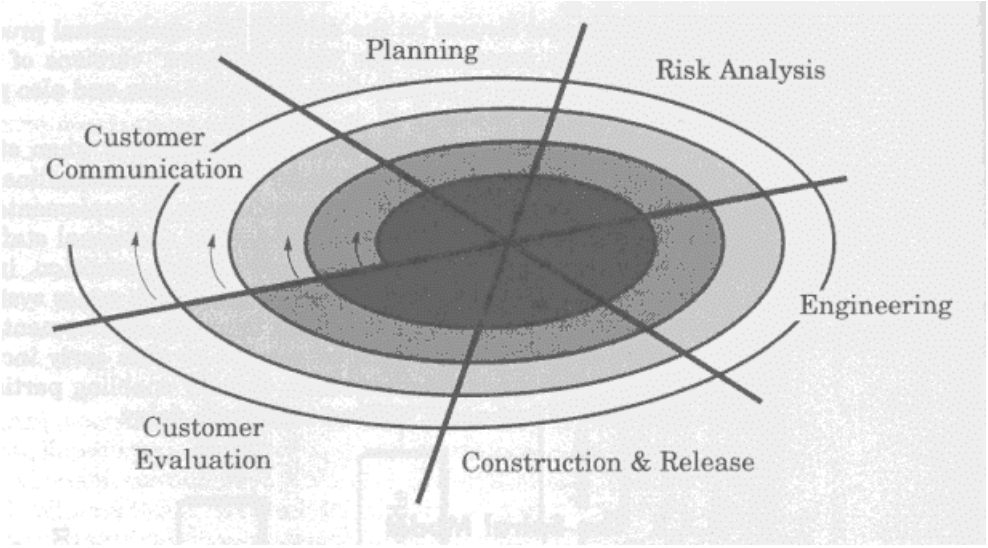
One mark should be awarded for each problem listed (up to a maximum of three marks). Examples include the following:

- **Schedule and cost estimates are often grossly inaccurate.**
- **The productivity of software developers has not kept pace with the demand for their services.**
- **The quality of software is sometimes less than adequate.**
- **There is little data on the software development process, forestalling improvements that can be made by looking at past experiences.**
- **Poorly-defined customer requirements at the start of a software development cycle has resulted in customer dissatisfaction with the completed software at the end of the cycle.**

(c) "The *spiral* model of development is said to combine aspects of prototyping and also the classic model." How does the spiral model combine prototyping and the linear sequential model? [2]

The spiral model combines the iterative nature of prototyping (that allows for the refinement of expectations before actual product construction) (1 mark) with the controlled and systematic aspects of the linear sequential model (1 mark).

(d) Draw a diagram showing the different steps of software development for the spiral model. In addition, describe each step of the model. [8]



One mark should be awarded for the correct placement of stages in the above and one mark should be awarded for the spiral-like representation.

- **Customer communication: tasks required to establish effective communication between developer and customer (1 mark).**
- **Planning: tasks required to define resources, timelines, and other project related information (1 mark).**
- **Risk analysis: tasks required to assess both technical and management risks (1 mark).**
- **Engineering: tasks required to build one or more representations of the application (1 mark).**
- **Construction and release: tasks required to construct, test, install and provide user support (e.g. documentation and training) (1 mark).**
- **Customer evaluation: tasks required to obtain customer feedback based on evaluation of the software representations created during the engineering stage and implemented during the installation stage (1 mark).**

Question 3

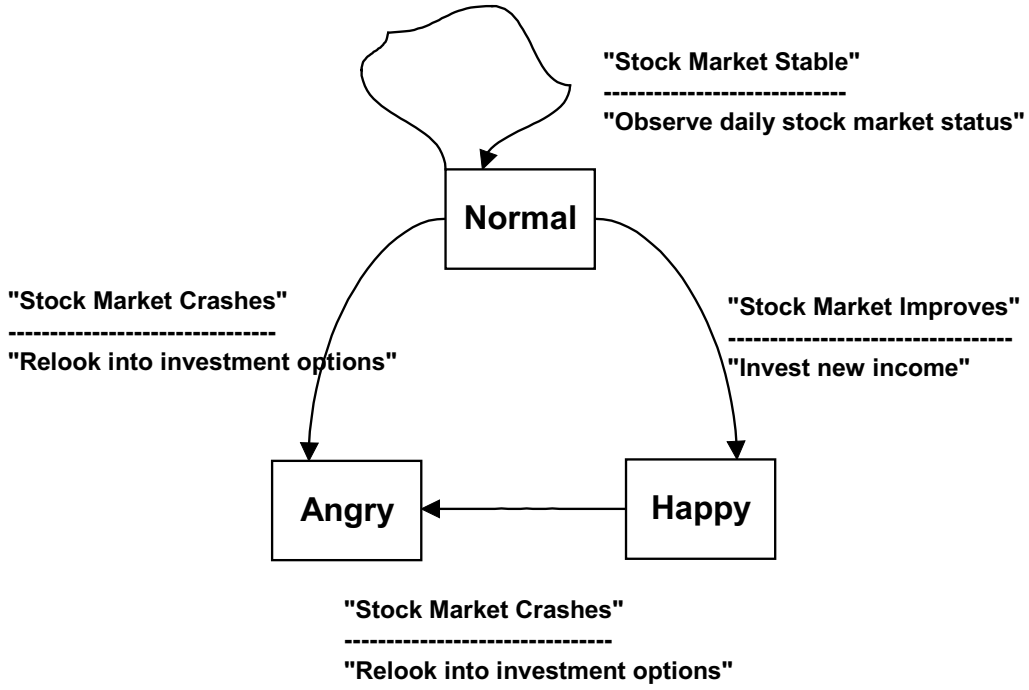
(a) Describe the three views of data and control in the information domain. [6]

Information flow (1 mark): represents the manner in which data and control change as each moves through a system (1 mark). Information content (1 mark): represents the individual data and control items that comprise some larger item of information that is transformed by the software (1 mark). Information structure (1 mark): represents the internal organisation of various data and control items (1 mark).

(b) A stock trader exists in three states of behaviour: "normal", "happy" and "angry". The default state for this trader is in "normal". Based on the daily happenings of the stock market, this trader can transit between each of the three states of behaviour. Specifically, when the trader is in "normal" state:

- If the stock market is stable, the trader continues to remain in "normal" state, and continues to observe the daily stock market status.
- If the stock market crashes, then the trader will immediately transit to "angry" state, regardless of the current state he is in (whether "normal" or "happy"), and immediately re-look into his investment options.
- If the stock market improves, the trader will transit from "normal" state to "happy" state, and look into ways of investing his income.

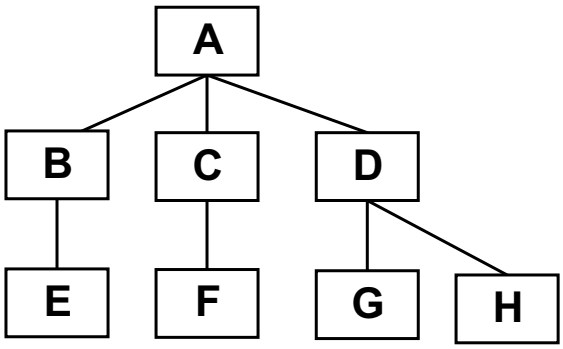
Translate this scenario into a simple state-transition diagram. [9]



Each of the three remaining transition arrows - 1 mark = 3 marks total
Each "condition" or "event" triggering the transition - 1 mark = 3 marks total
Each "consequence" or "action" that takes place as a result - 1 mark = 3 marks total

Question 4

(a) Consider the following structure.



(i) In the context of the above structure, what do we mean by *depth*?

(ii) In the context of the above structure, what do we mean by *width*?

(iii) What is the width of this structure?

(iv) What is the relationship between module C and module F?

(v) What is the *fan out* of module D?

(vi) What is the size of this structure? [7]

(i) The depth is the number of detail levels in the structure (1 mark).

(ii) The width is the overall span of control (1 mark).

(iii) The width is 4 (1 mark).

(iv) Module C is superordinate to module F (or module F is subordinate to module C) (1 mark).

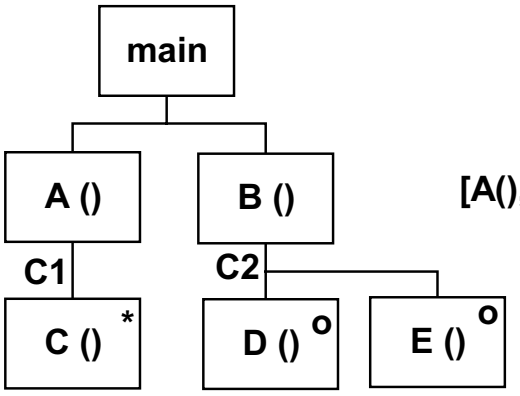
(v) The fan out of module D is 2 (1 mark).

(vi) Size = number of nodes + number of arcs (1 mark)
= 15 (1 mark)

(b) Consider the following simple pseudo-code fragment.

```
main {
      run subroutine A()
      run subroutine B()
}

A() {
      while condition C1 is true, run subroutine C()
}

B() {
      If condition C2 is true, run subroutine D();
      else run subroutine E()
}

C() {
      ADD numbers
}

D() {
      SUBTRACT numbers
}

E() {
      DIVIDE numbers
}
```

Translate this code fragment into a structure using Jackson's notation. [8]



[A(), B() - 1 mark each = 2 marks]

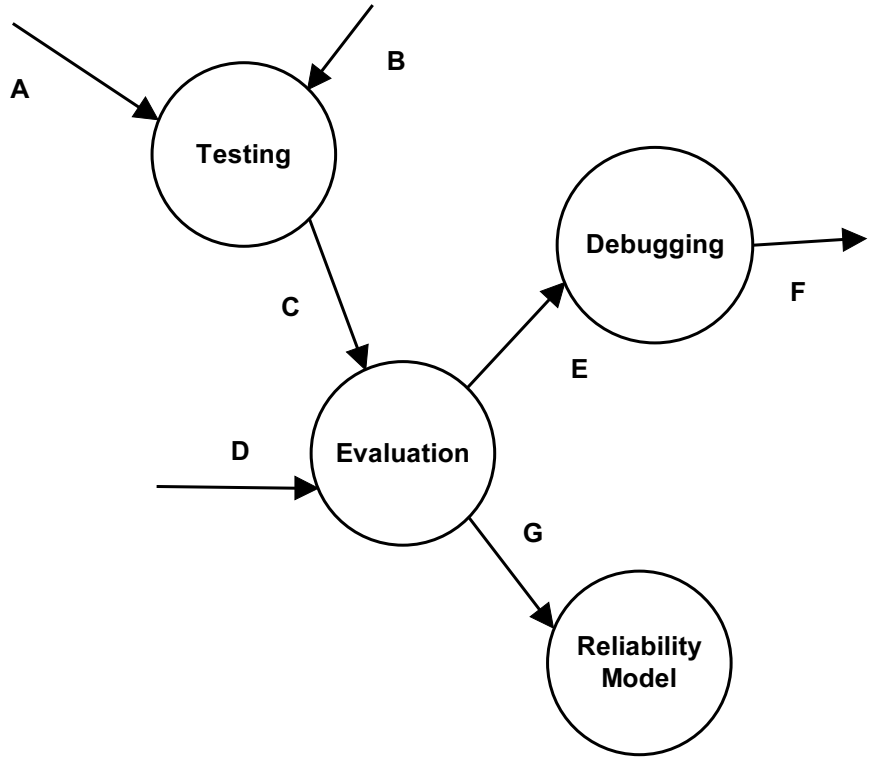
[C1 and C2 placement - 1 mark each = 2 marks]

[C() - 1 mark, iterative symbol - 1 mark. Total = 2 marks]

[D(), E() - 1 mark, selection symbol - 1 mark. Total = 2 marks]

Question 5

(a) The following diagram demonstrates testing in relation to other processes. Based on your understanding of testing as an activity in software development, describe each process in relation to testing. In your answer, also your clarify what items A to G are, and how they are used in each process. [8]



A = Software configuration; B = Hardware configuration; C = Test Results; D = Expected Results; E = Errors; F = Corrections to errors; G = Error rate data

The following mark allocation is to be used:
Description for each of the four processes (1 mark each, total 4 marks) – see page 8-2.
Data flows (A to G): 1 mark for one or two correct; 2 marks for three or four correct; 3 marks for between 5 or 6 correct; 4 marks for 7 correct.

(b) All software testing strategies are said to share common characteristics. List *three* of these characteristics. [3]

One mark should be awarded for each characteristic named (up to a maximum of three marks). The characteristics are as follows.

- **Testing beings at the module level and works "outward" toward the integration of the entire computer-based system.**
- **Different testing techniques are appropriate at different points in time.**
- **Testing is conducted by the developer of the software and (for large projects) an independent test group.**
- **Testing and debugging are different activities, but debugging must be accommodated in any testing strategy.**

(c) Describe each of the following types of system testing.

(i) Recovery testing.

(ii) Security testing.

(iii) Stress testing.

(iv) Performance testing. [4]

(i) A system test that forces software to fail in a variety of ways and verifies that recovery is properly performed (1 mark).

(ii) Security testing attempts to verify that protection mechanisms built into a system will in fact protect it from improper penetration (1 mark).

(iii) Stress testing is designed to confront programs with abnormal situations where unusual quantity frequency, or volume of resources are demanded (1 mark).

(iv) Performance testing seeks to test the run-time performance of software within the context of an integrated system (1 mark).

-END OF PAPER-