5. a) Why is software difficult to design?

[10 marks]

b) How can management organisation and the systematic application of methods increase the chance that a development project will succeed?

[10 marks]

b) What are the features of good software design? Can quality be measured?

[13 marks]

[Total 33 marks]

END OF PAPER

3. Development methods involve building models to describe the system being investigated.

a) What are the three kinds of model developed during object-oriented analysis and design?

[6 marks]

b) How are these different models made consistent?

[10 marks]

c) What kind of model is built using formal specification? What are the key features that distinguish formal specification from object-oriented development?

[8 marks]

d) It could be said that lines of source code give the only model that matters. What are the flaws in this argument?

[9 marks]

[Total 33 marks]

4. a) What are design patterns and what are their key benefits?

[10 marks]

b) Explain the idea of reuse. What can be reused and how does reuse affect the development process?

[16 marks]

c) What features of object-oriented design and programming enable and support reuse?

[7 marks]

[Total 33 marks]

CONTINUED

Answer THREE Questions

1. Consider the following specification for a software system:
   "A system is required to store text documents on-line. When a document is entered into the system it should include a title, author list and abstract. Each document is dated and has an owner who is the only person allowed to edit or delete it. To allow for rapid retrieval, documents are automatically indexed by every word occurring in the document. The user may search for documents by title, author, or by a general keyword search."

   a) Generate a list of potential object classes and attributes, briefly explaining why each was selected.

   [10 marks]

   b) Draw a class diagram using the classes that have been identified.

   [10 marks]

   c) Write a short technical report on the feasibility of developing a system based on the specification, including a summary of how to proceed with further development.

   [13 marks]

   [Total 33 marks]

2. The following 11 terms are commonly used in a software engineering context. Give a short explanation for each of them:
   abstraction, encapsulation, CASE tool, structure chart, module,
   top-down decomposition, cohesion, metrics, pseudocode, generic, framework

   [3 marks each]

   [Total 33 marks]