B3.3-R3: SOFTWARE ENGINEERING & CASE TOOLS

NOTE:

1. Answer question 1 and any FOUR questions from 2 to 7.
2. Parts of the same question should be answered together and in the same sequence.

Time: 3 Hours **Total Marks: 100**

1. Give brief answers to the following questions.

a) Briefly, explain the problems that may arise if the build and fix model is used for developing a large software product.

b) What is the difference between functional testing and structural testing? Can system testing be considered to be a structural testing? Explain your answer.

c) Do you agree with the claim: "The essence of any good function-oriented design technique is to map the functions performing similar activities into a module." Justify your answer.

d) What is the difference between object-oriented analysis and object-oriented design? How are the oucomes of these processes documented?

e) What do you understand by structural complexity of a program? How can it be measured?

f) Besides code commenting, explain the different ways in which a program source code can be documented.

g) Do you agree with the statement: "One of the objectives of the system testing is to check whether coding standards have been adhered to or not." Justify your answer.

(7x4)

2.

a) What are the advantages and disadvantages of formal specification over traditional specification? Give at least one example for which formal specification is desirable and another for which formal specification is undesirable.

b) List the important shortcomings of LOC for use as a software size metric. Does the function point metric overcome these? Explain your answer.

c) Do you agree with the following statement: "Modern quality assurance paradigms are centered around carrying out through product testing." Justify your answer.

(6+6+6)

3.

a) Explain, how the reusability of a developed software module can be enhanced.

b) Discuss the role of the data dictionary in a CASE environment. How auotomated support for data dictionary can be provided?

c) Discuss the process models that can be used for software maintenance and indicate, how you would select an appropriate process model for a maintenance project at hand.

(5+6+7)

4.

a) What do you understand by performance testing? What are the different types of performance testing that are normally carried out?

b) Explain the importance of software configuration management in modern quality paradigms such as SEI CMM and ISO 9001. What problems might arise if a development organization does not use any configuration management tool?

c) What do you understand by software re-engineering? How is it different from reverse engineering? Explain situations under which each would be useful.

(6+6+6)

5.

a) What do you understand by unit testing? Explain, how you can design black-box test cases for a function named **quadratic-eqn-solver**. Quadratic-eqn-solver accepts three floating point numbers representing a quadratic equation of the form $ax^2+bx+c=0$, it computes and displays the solution.

b) What do you understand by requirements validation? How can requirements be validated?

c) Explain the metrics that you would use to measure the following factors of an object-oriented program also briefly mention, how you would estimate these characteristics:

- design quality
- estimated number of faults
- maintainability
- estimated test effort
- estimated program development effort

(6+6+6)

6.

a) Suppose a very large software is to be developed for an innovative application. Naturally, the development would be beset with many risks due to the innovative product to be developed. Explain which development model should be used. Explain how the risks would be handled in the model.

b) What do you mean by process modelling? Why is it required to model processes?

c) Explain the human cognition capabilities (relevant to human-computer interfaces) and their limitations. How do these influence design of effective human-computer interfaces?

(5+6+7)

7.

a) Do you think that testing object-oriented programs is easier than testing procedural programs? Explain your answer with special mention as to how the object-orientation featuers of inheritance, encapsulation, polymorphism, and dynamic binding influence effective test case design.

b) Explain why reuse of code has not been very successful. In this context, explain, why design patterns can be considered to be an effective form of software reuse.

c) Explain static and dynamic tool support that can be provided during program testing. What are the different types of results produced by these tools? Explain, how these are useful in testing.

(6+6+6)