

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.

- a) What do you mean by software crisis? What are some of its indicators?
- b) What do you mean by the syntactic domain and the semantic domain of a formal specification language?
- c) What is software reengineering? Why is it required?
- d) What do you mean by change control? What is its need?
- e) What are the different tools used in a CASE environment? How are these tools integrated?
- f) Differentiate between verification and validation of a software product. When would you carry them out?
- g) What is the difference between error and failure in the context of software testing? Which one would you detect while testing?

(7x4)

2.

- a) Bring out the difference between software requirements analysis and software requirements validation.
- b) What are the objectives of software requirements validation? What are some inputs and outputs of the software requirements validation process?
- c) Define Coupling in the context of software design. What are the different types of coupling in practice? Discuss them briefly. Also discuss why data coupling is the best form of coupling.

(4+6+8)

3.

- a) What do you mean by a CASE tool and a CASE environment? Why integration of the CASE tools is a necessity?
- b) What are the primary objectives of developing CASE tools? What are the different facilities that a CASE environment provides?
- c) Does the use of CASE tools for software development help in cost saving? Why and how much?

(6+6+6)

4.

- a) What is a test case? How is it different from a test suite?
- b) What do you mean by Black Box Testing? What are some considerations in this regard? Discuss two methods of black box testing in detail.
- c) In an automatic banking operation, the users need to enter an area code that should assume values between 200 and 999. Find the test cases for the area code for the relevant program from considerations of:
 - Equivalence class partitioning
 - Boundary value analysis

(4+8+6)

5.

- a) Define software quality. Define the following software product quality factors – portability, consistency, maintainability, testability, usability, and reliability.
 - b) Bring out the objectives of a formal technical review for software quality assurance.
 - c) What are some common software defects that may occur in the context of software development? Discuss them briefly. Which of these defects are usually more prominent?
- (8+4+6)**

6.

- a) In spite of the fact that abstract classes cannot be instantiated, why are they used? Discuss with an example.
 - b) What are Class diagrams? What do they contain? Give an example.
 - c) What do you mean by domain analysis in object-oriented software engineering?
- (6+6+6)**

7.

- a) What is the cleanroom approach to software engineering? What are its special features? What are the benefits of using the cleanroom approach?
 - b) Define the Component-based software engineering process. What are its steps? What are its advantages and disadvantages?
- (10+8)**