

Section A

Q.-1. Answer the following (Attempt any three) (30)

- (1) Explain the components of an ERD and discuss the three types of cardinalities (1:1, 1:M, M:N) with relevant examples.
- (2) What are the characteristics of a good Software Requirement Specification (SRS) document? Provide the standard template/format for writing an SRS.
- (3) Describe the various tools and techniques used in requirement gathering, such as Review of Records, On-site Observation, Questionnaires, and Interviews.
- (4) Compare and contrast the Classical Waterfall Model and the Iterative Waterfall Model, discussing their structure, advantages, and practical limitations.
- (5) Provide a detailed overview of the various phases of the Software Development Life Cycle (SDLC), describing the primary tasks performed in each stage from Feasibility to Maintenance.

Section B

Q.-2. Write short notes on any four of the following topics. (20)

- (1) Summarize the features of the Evolutionary Model and when it is considered suitable for project delivery.
- (2) Describe the Delphi technique for software estimation and how it improves upon single-expert judgment.
- (3) Briefly compare Democratic, Chief Programmer, and Mixed team structures in a software organization.
- (4) Identify and briefly describe the elements of a Use-Case Diagram, specifically Actors and System Boundaries.
- (5) Explain the role of Lifelines and Activation boxes in a UML Sequence Diagram.
- (6) Explain the Robustness Testing method in Boundary Value Analysis and how it differs from standard BVC.

Section C

Q.-3. Choose the correct option (A, B, C, or D). (10)

(1) Which feasibility study ensures the cost incurred is less than the benefits served?

- | | |
|--------------|----------------|
| A) Technical | B) Operational |
| C) Economic | D) Schedule |

(2) The outcome of the Requirement Gathering, Analysis and Specification phase is:

- | | |
|----------------|----------------|
| A) SDD | B) SRS |
| C) Source Code | D) Test Report |

(3) How many factors are considered when calculating the Technical Complexity Factor (TCF)?

- | | |
|-------|-------|
| A) 5 | B) 7 |
| C) 10 | D) 14 |

(4) In which team structure is there no formal hierarchy, leading to higher job satisfaction?

- | | |
|---------------------|----------------|
| A) Chief Programmer | B) Mixed |
| C) Democratic | D) Centralized |

(5) In a Data Flow Diagram (DFD), a circle represents a:

- | | |
|------------|---------------|
| A) Entity | B) Data Store |
| C) Process | D) Data Flow |

(6) A key column in a data table that uniquely identifies each row and cannot be null is a:

- | | |
|----------------|------------------|
| A) Foreign Key | B) Composite Key |
| C) Primary Key | D) Secondary Key |

(7) In a Use-Case diagram, a user or external role interacting with the system is an:

- A) Instance
- B) Use-case
- C) Actor
- D) Object

(8) The thumb rule for good software modular design is:

- A) High Coupling, Low Cohesion
- B) High Cohesion, Low Coupling
- C) Low Cohesion, Low Coupling
- D) High Cohesion, High Coupling

(9) In Boundary Value Analysis (BVC), how many test cases are designed for 'n' variables?

- A) $2n+1$
- B) $4n+1$
- C) $6n+1$
- D) n^2

(10) A dummy module used in top-down integration testing when a lower-level module is not ready is a:

- A) Driver
- B) Stub
- C) Placeholder
- D) Anchor

Section D

Q.-4. Decide whether the statements given below are True or False. (10)

- (1) Software products are usually developed by individuals for their personal use.
- (2) Function point metrics can be used to estimate software size irrespective of the programming language.
- (3) In a democratic team structure, team members follow a formal hierarchy and report to a single central authority.
- (4) Explicit requirements are those stated by the customer, while implicit requirements are those necessary for the system to behave properly.
- (5) Encapsulation ensures that data in a class can only be accessed through its member methods.
- (6) Verification checks if the software is being built according to requirements, effectively asking "are we developing the right product?".
- (7) Static testing involves checking documents and code without actually executing the software.
- (8) Black-box testing requires the tester to have an in-depth understanding of the programming language used in the code.
- (9) Statement coverage is a white-box testing method that validates if every line of code is executed at least once.
- (10) Integration testing is the activity of testing the smallest building blocks of a system individually.
