Course Code	**:**	**BCS-151**
Course Title	**:**	**Introduction to Software Engineering**
Assignment Number	**:**	**BCA_NEW/BCAOL(V)151/Assign/2026-27**
Maximum Marks	**:**	**100**
Last Dates for Submission	**:**	**31st October, 2026 (For July 2026 Session)**
	:	**30th April, 2027 (For January 2027 Session)**

Note: This assignment has eight questions carrying 10 marks each for a total of 80 marks. Answer all questions. The remaining 20 marks are for Viva Voce. Draw suitable diagrams wherever necessary.

Q1:(a) Explain the characteristics of a **good Software Requirements Specification (SRS)**. Discuss any **five characteristics** recommended for preparing an effective SRS document. **(5 Marks)**

(b) A university intends to develop an **Online Examination Management System**. Identify **five functional requirements** and **five non-functional requirements** that should be included in its SRS. **(5 Marks)**

Q2: An organization plans to automate its **Library Management System**.

(a) Draw a **Context Diagram** showing the interaction between the system and external entities. **(4 Marks)**

(b) Develop a **Level-1 Data Flow Diagram (DFD)** consisting of at least **four major processes**. **(6 Marks)**

Q3: A company wishes to develop an **Online Grocery Delivery System** using an object-oriented approach.

(a) Identify the major classes, their attributes and operations. **(5 Marks)**

(b) Draw a **Use Case Diagram** showing the interaction among Customer, Delivery Executive and Administrator. **(5 Marks)**

Q4: A software company estimates that a project consists of **480 Function Points**.
The productivity of the development team is **12 Function Points per Person-Month.**

The project is expected to be completed in **8 months.**

(a) Calculate the estimated effort in person-months. **(3 Marks)**

(b) Estimate the required team size. **(3 Marks)**

(c) Discuss any **four advantages** of Function Point Analysis over LOC-based estimation. **(4 Marks)**

Q5:(a) Explain the concept of **Cohesion** and **Coupling**. Illustrate different types of Cohesion and Coupling with suitable examples. **(5 Marks)**

(b) Why are **High Cohesion** and **Low Coupling** considered desirable in software design? **(5 Marks)**

Q6: An e-Commerce company has completed the development of its shopping portal.

(a) Prepare **four Black Box Test Cases** for the Login Module. **(5 Marks)**

(b) Differentiate between: **(5 Marks)**

- Verification and Validation
- Alpha Testing and Beta Testing

with suitable examples.

Q7:(a) Describe the major activities involved **in Software Project Management**. **(5 Marks)**

(b) Explain how **Risk Management** contributes to the successful completion of software projects. Illustrate the software risk management process with a suitable diagram. **(5 Marks)**

Q8:(a) Compare any **five Software Development Life Cycle (SDLC) Models** based on: **(5 Marks)**

- Requirement Stability
- Risk Handling
- Customer Participation
- Cost
- Suitable Applications

(b) Discuss the role of the following in Software Engineering: **(5 Marks)**

- Software Configuration Management (SCM)
- Software Maintenance
- Software Quality Assurance (SQA)

Explain how these activities collectively improve software quality.

Q1:(a) Explain the characteristics of a good Software Requirements Specification (SRS). Discuss any five characteristics recommended for preparing an effective SRS document. (5 Marks)

(b) A university intends to develop an Online Examination Management System. Identify five functional requirements and five non-functional requirements that should be included in its SRS. (5 Marks)

Answer

(a) Characteristics of a good SRS

A Software Requirements Specification (SRS) is a written agreement describing what a proposed software system must do, the conditions under which it must operate, and the constraints it must satisfy. It gives developers, testers, project managers and the customer a common basis for planning and checking the final product. A useful SRS describes observable requirements rather than prescribing unnecessary internal design decisions.

Correctness: Every stated requirement must reflect a genuine stakeholder need. For example, if university rules allow a candidate to download an admit card only after fee clearance, the SRS should state that rule accurately. Reviewing requirements with examination officials helps remove misunderstandings.

Unambiguity: Each requirement should have one clear interpretation. The statement “the portal should be fast” is vague. “The result page shall respond within three seconds for 95% of requests under 2,000 concurrent users” is more precise and can be tested.

Completeness: The SRS should cover major functions, input validation, performance, security, error situations, external interfaces and important operating constraints. It must not ignore what happens if a payment fails, the network disconnects, or a student enters an invalid registration number.

Consistency: Requirements must not contradict each other. A rule permitting only one final submission cannot coexist with another permitting unlimited post-deadline editing unless the exception is explained. Consistency also concerns names, units and terminology throughout the document.

Verifiability: There must be a practical way to establish whether each requirement is met through a test, review, demonstration or measurement. “The system shall lock the examination after the allotted duration” is verifiable using a timed test; “the system shall be user-friendly” needs measurable criteria.

Other useful qualities are traceability (a unique requirement ID links the need to design and tests), modifiability (changes can be made without creating confusion), and priority/ranking (essential requirements are distinguished from desirable