Criteria	Excellent (8-10 marks)	Good (5-7 marks)
(a) Explanation of Coupling and Cohesion	Clearly and accurately defines module coupling and cohesion. Provides clear and relevant examples of high/low coupling and high/low cohesion.	Defines module coupling and cohesion with minor inaccuracies. Provides examples with some relevance to high/low coupling and high/low cohesion.
(b) Benefits of High Cohesion and Low Coupling	Provides a comprehensive discussion of the benefits, including improved maintainability, reusability, testability, and reduced complexity.	Discusses some benefits of high cohesion and low coupling, but the explanation may lack depth or clarity.
(c) Online Store System Design	Identifies three relevant modules for an online store and provides a clear and logical explanation of how to achieve low coupling and high cohesion between them.	Identifies three modules, but the explanation of how to achieve low coupling and high cohesion may be incomplete or lack clarity.

Criteria	Excellent (8-10 marks)	Good (5-7 marks)
(a) Importance and Techniques of Defensive Coding	Clearly explains the importance of defensive coding in preventing errors and improving software robustness. Provides three relevant and accurate examples of defensive coding techniques (e.g.,	Explains the importance of defensive coding with some clarity. Provides some relevant examples of defensive coding techniques, but may lack detail or accuracy.
(b) Stages and Benefits of User Testing	Provides a comprehensive description of the different stages involved in user testing (e.g., planning, recruitment, execution, analysis). Clearly explains how user testing results can improve	Describes some stages of user testing, but the explanation may lack detail or clarity. Explains how user testing can improve design with some limitations.

(c) Defensive Coding Implementation	Writes a function with clear and effective defensive coding techniques to handle invalid inputs. The code is correct, efficient, and well-structured.	Writes a function that incorporates some defensive coding techniques, but may miss some potential invalid input scenarios or have minor errors.

Criteria	**Excellent (8-10 marks)**	**Good (5-7 marks)**
(a) Benefits and Concepts of Version Control	Clearly explains the benefits of using version control systems (e.g., collaboration, tracking changes, rollback). Provides a clear and accurate description of branching, merging, and	Explains some benefits of version control, but the explanation may lack detail or clarity. Describes branching, merging, and committing with some limitations.
(b) Process and Advantages of TDD	Provides a comprehensive outline of the TDD process (Red-Green-Refactor). Clearly discusses the advantages of using TDD (e.g., reduced bugs, improved design, increased confidence).	Outlines the TDD process with some clarity. Discusses some advantages of TDD, but the explanation may lack depth.
(c) TDD Implementation for Email Validation	Clearly describes the steps involved in developing the email validation function using TDD. Provides clear and relevant examples of tests (e.g., valid email format, invalid email	Describes the TDD steps with some clarity. Provides some examples of tests and code, but may lack detail or accuracy.

Satisfactory (3-4 marks)	Needs Improvement (1-2 marks)
Definitions of module coupling and cohesion lack clarity or accuracy. Examples are vague or not clearly linked to the concepts.	Fails to define module coupling and cohesion or provides inaccurate definitions. Examples are missing or irrelevant.
Mentions a few benefits of high cohesion and low coupling, but the discussion is superficial or incomplete.	Fails to discuss the benefits of high cohesion and low coupling or provides irrelevant information.
Identifies modules with limited relevance to an online store. Struggles to explain how to achieve low coupling and high cohesion.	Fails to identify relevant modules or provides a vague and inadequate explanation of low coupling and high cohesion.

Satisfactory (3-4 marks)	Needs Improvement (1-2 marks)
Provides a basic explanation of the importance of defensive coding. Examples of defensive coding techniques are vague or incomplete.	Fails to explain the importance of defensive coding or provides irrelevant information. Examples are missing or inaccurate.
Provides a basic overview of user testing stages. The connection between user testing results and design improvements is weak.	Fails to describe the stages of user testing or provides inaccurate information. Does not explain how user testing improves design.

Writes a function with limited defensive coding techniques. The code may contain errors or inefficiencies.	Fails to write a function or the code lacks any defensive coding measures.

Satisfactory (3-4 marks)	**Needs Improvement (1-2 marks)**
Provides a basic explanation of version control and its benefits. Definitions of branching, merging, and committing are vague or incomplete.	Fails to explain the benefits of version control or provides inaccurate information. Struggles to describe branching, merging, and committing.
Provides a basic overview of the TDD process. The discussion of advantages is superficial or incomplete.	Fails to outline the TDD process or provides inaccurate information. Does not discuss the advantages of TDD.
Provides a basic description of the TDD steps for email validation. Examples of tests and code are vague or incomplete.	Fails to describe the TDD steps or provides inaccurate information. Examples are missing or irrelevant.