

SRM VALLIAMMAI ENGINEERING COLLEGE

SRM Nagar, Kattankulathur – 603 203

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

QUESTION BANK (Common to B.Tech - AI&DS)



III SEMESTER

CS3363 – SOFTWARE ENGINEERING

Regulation – 2023

Academic Year 2026 – 2027 ODD

Prepared by

Dr. S. Venkatesh – Assistant Professor(Sr.G) / CSE

Ms.R.Gayathri, Assistant Professor (O.G)/CSE

Ms.M.Sophiya Sugantha Grace – Assistant Professor (O.G) /CSE

Ms.R.Manjupriya, Assistant Professor (O.G)/AI-DS

Mr.B.Yogesh Kumar, Assistant Professor (O.G)/AI-DS



SRM VALLIAMMAI ENGINEERING COLLEGE
SRM Nagar, Kattankulathur – 603203.



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
QUESTION BANK

SUBJECT : CS3363- SOFTWARE ENGINEERING

SEM / YEAR : III / II

UNIT I - SOFTWARE PROCESS AND AGILE DEVELOPMENT			
Introduction to Software Engineering, Software Process, Perspective and Specialized Process Models – Introduction to Agility-Agile process-Extreme programming-XP Process - Quality management-SQA-SQA plan.			
PART-A (2 - MARKS)			
Q. No	QUESTIONS	Competence	BT Level
1.	Write the IEEE definition of software engineering.	Remember	BTL-1
2.	List the umbrella activities of the software process.	Remember	BTL-1
3.	Define the goals of software engineering.	Remember	BTL-1
4.	Differentiate between verification and validation. Give an example.	Understand	BTL-2
5.	List the characteristics of software contrasting it with characteristics of hardware.	Remember	BTL-1
6.	What is a software process that manages unpredictability?	Understand	BTL-2
7.	Identify the human factors considered in agile software development.	Understand	BTL-2
8.	What is the Win-Win Spiral Model?	Understand	BTL-2
9.	Mention any two advantages and two disadvantages of the iterative software development model.	Understand	BTL-2
10.	Define the agile process. Give any two agile principles.	Remember	BTL-1
11.	List any two deficiencies of the waterfall model.	Remember	BTL-1
12.	Differentiate between prescriptive and specialized process models.	Understand	BTL-2
13.	What is an XP Story?	Understand	BTL-2
14.	Mention any two drawbacks of the spiral model.	Understand	BTL-2
15.	Mention any two characteristics of software as a product.	Understand	BTL-2
16.	Why did software development shift from product-oriented to process-oriented development?	Understand	BTL-2
17.	Differentiate between SDD and DDD.	Understand	BTL-2
18.	Write the generic process framework applicable to any software project.	Remember	BTL-1

19.	What is Extreme Programming (XP)?	Understand	BTL-2
20.	Why should system engineers understand the environment of a system? Give two reasons.	Understand	BTL-2
	What are the potential advantages of software life cycle models?	Understand	BTL-2
21.	Differentiate between object-oriented and function-oriented software design.	Understand	BTL-2
22.	What are the umbrella activities of a software process?	Understand	BTL-2
23.	List any two deficiencies of the waterfall model.	Remember	BTL-1
24.	Define software process improvement (SPI).	Remember	BTL-1
PART-B (16- MARKS)			

1.	Illustrate the Spiral Model with a neat diagram and explain how it is applied in the software life cycle. List the different software life cycle models.	(16)	Apply	BTL-3
2.	(i) Explain atleast one scenario where a) RAD model would be applicable and not the waterfall model. b) waterfall model is preferable compare to all other models. (ii) What are the pros and cons of using mathematical approach for software development?	(8) (8)	Analyze	BTL-4
3.	(i) Describe about agile modeling in detail. (ii) Explain the component based software development model with a neat sketch	(8) (8)	Analyze	BTL-4
4.	(i) Write short notes on aspect oriented software development. (ii) Explain in detail about personal process models and team process models.	(8) (8)	Evaluate	BTL-5
5.	(i) What is a process model? Describe the process model that you would choose to manufacture a car explain giving suitable reasons. (ii) Describe the various Evolutionary Process Models with neat diagram.	(8) (8)	Analyze	BTL-4
6.	(i) Compare the life cycle models based on their distinguishing factors, strengths and weaknesses. (ii) Discuss the prototyping model .what is the effect of designing prototype on the overall cost of the software project?	(8) (8)	Evaluate	BTL-5

7.	Explain in detail about iterative and waterfall model and also write short notes on concurrent models.	(16)	Analyze	BTL-4
8.	(i) Discuss in detail about drawback of life cycle model. (ii) What is the significance of the spiral model when compared with other model?	(8) (8)	Analyze	BTL-4
9.	Apply the practices of Extreme Programming (XP) to a software development project and explain the issues that lead to the XP debate.	(16)	Apply	BTL-3
10.	(i) Illustrate about agility and cost of change. (ii) What key traits must exist among the people on an effective software team?	(8) (8)	Apply	BTL-3
11.	Evaluate the importance of agility in software engineering and justify how the principles of agile software development contribute to successful software projects.	(16)	Evaluate	BTL-5
12.	(i) Compose your view about agile software development. (ii) Generalize your view about extreme programming.	(8) (8)	Create	BTL-6
13.	Describe about pair programming and how unit tests used in XP? And list the new practices appended to XP to create IXP.	(16)	Analyze	BTL-4
14.	(i) Explain Software product engineering with its services and advantages. (ii) Write a note on the unique characters of a software.	(8) (8)	Apply	BTL-3
15.	Which process model is best suited for risk management? Discuss in detail with an example. Give the advantages and disadvantages of the model.	(16)	Analyze	BTL-4
16.	Describe the pair XP concept of refactoring and programming.	(16)	Analyze	BTL-4
17.	Analyze the concept of Agility. List the principles of agility and illustrate the process in detail.	(16)	Analyze	BTL-4

UNIT II- REQUIREMENTS ANALYSIS AND SPECIFICATION

Software Requirements: Functional and Non-Functional, User requirements, System requirements, Software Requirements Document – Requirement Engineering Process: Feasibility Studies, Requirements elicitation and analysis, requirements validation, requirements management-Classical analysis: Structured system Analysis, Petri Nets- Data Dictionary.

PART-A (2 - MARKS)

Q.No	QUESTIONS	BT Level	Competence
1.	State the use cases involved in an online shopping system with customer registration, authentication, and online payment through a payment gateway such as PayPal.	Remember	BTL-1

2.	Define feasibility study. And list the types.	Remember	BTL-1	
3.	Identify the non-functional requirements of a banking system.	Understand	BTL-2	
4.	List the entities in a simple semantic data model for a Library Management System.	Remember	BTL-1	
5.	List the characteristics of a good system requirements specification(SRS)	Remember	BTL-1	
6.	Define Quality Function Development(QFD)	Remember	BTL-1	
7.	List the classifications of requirements with an example for each.	Remember	BTL-1	
8.	Identify the phases of the spiral view of the requirements engineering process.	Remember	BTL-1	
9.	Differentiate between normal and exciting requirement.	Understand	BTL-2	
10.	List the problems faced when user requirements are written in natural language.	Remember	BTL-1	
11.	Distinguish between the terms inception, elicitation and elaboration with reference to requirements.	Understand	BTL-2	
12.	List two advantages of using traceability tables in the requirements management phase.	Remember	BTL-1	
13.	List the metrics used for specifying non-functional requirements.	Remember	BTL-1	
14.	Express the different types of check list that should be carried out for requirement validation process.	Understand	BTL-2	
15.	State the steps involved in managing changing requirements during the requirements elicitation process.	Remember	BTL-1	
16.	What is meant by structural analysis and volatile requirement?	Remember	BTL-1	
17.	List the common data dictionary notations.			
18.	Define Petri Net and list types of traceability in a software process.	Remember	BTL-1	
19.	State the requirements validation techniques.	Remember	BTL-1	
20.	Define the concept of a data dictionary.	Remember	BTL-1	
21.	What are the types of prototypes?	Remember	BTL-1	
22.	Interpret the usage of ERD.	Understand	BTL-2	
23.	List the advantages of using traceability tables in requirements management.	Remember	BTL-1	
24.	Identify the relationship between Data Flow Diagrams (DFD) and Entity–Relationship (E-R) Diagrams.	Remember	BTL-1	
PART-B (16- MARKS)				
1.	Analyze the differences between functional and non-functional requirements and examine the steps involved in initiating the requirements engineering process.	(16)	Analyze	BTL-4

2.	(i) Apply the concept of non-functional requirements to a software system and illustrate them with suitable examples. (ii) Apply the concepts of user requirements and system requirements by distinguishing them with suitable examples.	(8) (8)	Apply	BTL-3
3.	Evaluate the importance of documenting the three aspects of an SRS and justify the characteristics and components of a good SRS document.	(16)	Evaluate	BTL-5
4.	(i) Construct the structure of a requirements document for a software project. (ii) Demonstrate how different stakeholders use a requirements document.	(8) (8)	Apply	BTL-3
5.	(i) Analyze the different methods used to write a system requirements specification. (ii) Analyze the spiral view of system requirements engineering.	(8) (8)	Analyze	BTL-4
6.	Analyze the requirements engineering process and examine how requirements are managed throughout the software development life cycle.	(16)	Analyze	BTL-4
7.	Evaluate the feasibility study process by assessing its purpose, inputs, outputs issues addressed, and phases involved.	(16)	Evaluate	BTL-5
8.	Evaluate the requirements elicitation process by assessing the activities involved using a watch system for setting time and alarms as an example.	(16)	Evaluate	BTL-5
9.	(i) Design a feasibility study framework for a software project and illustrate how it supports requirements engineering. (ii) Develop a requirement classification model and identify stakeholders and requirement types for an online train reservation system.	(8) (8)	Create	BTL-6
10.	Write short notes on the list given below (i) Requirements discovery and Interviewing. (ii) Scenarios and Use cases.	(8) (8)	Remember	BTL-1
11.	(i) Classify the different types of checks carried out on the requirements in the requirements document during the validation process. (ii) Demonstrate on the requirement validation techniques.	(8) (8)	Apply	BTL-3
12.	(i) Analyze the activities involved in requirements management planning. (ii) Analyze the process of requirements change management.	(8) (8)	Analyze	BTL-4
13.	(i) Evaluate the effectiveness of structural system analysis in software development. (ii) Assess the classical Petri Nets model and its significance.	(8) (8)	Evaluate	BTL-5

14.	Design and construct Context, Level-0, and Level-1 Data Flow Diagrams for a Library Management System and incorporate appropriate DFD notations.	(16)	Create	BTL-6
15.	Analyze the functional and behavioral models used in the software requirements process and compare their roles.	(16)	Analyze	BTL-4
16.	Draw use case & data flow diagrams for a “restaurant system”. The activities of the Restaurant system are listed below. Receive the customer food orders, Produce the customer ordered foods, Serve the customer with their ordered foods, collect payment from customers, store customer payment details, order raw materials for food products, pay for raw materials & pay for labor.	(16)	Apply	BTL-3
17.	Evaluate the differences between an SRS document and a Design document, and justify the essential contents of each document.	(16)	Evaluate	BTL-5

UNIT III- SOFTWARE DESIGN

Design process – Design Concepts-Design Model– Design Heuristic – Architectural Design -Architectural styles, Architectural Design, Architectural Mapping using Data Flow- User Interface Design: Interface analysis,Interface Design – Component level Design: Designing Class based components, traditional Components.

PART-A (2 - MARKS)

1.	What do you interpret from design heuristics?	Understand	BTL-2
2.	List two principles of good design.	Remember	BTL-1
3.	What do you infer from the design quality attributes ‘FURPS’?	Remember	BTL-1
4.	Draw the context flow graph of an ATM automation system.	Remember	BTL-1
5.	State the characteristics of a loosely coupled and highly cohesive system.	Remember	BTL-1
6.	Define Modularity.	Remember	BTL-1
7.	Give the various types of architectural styles with example.	Understand	BTL-2
8.	What is coupling and list the various types of coupling?	Remember	BTL-1
9.	How do you apply modularization criteria for monolithic software? Discuss.	Understand	BTL-2
10.	Define mapping.	Remember	BTL-1
11.	List the UI design patterns used in software applications.	Remember	BTL-1
12.	Distinguish between transform flow and transaction flow.	Understand	BTL-2
13.	List the basic design principles of class based component.	Remember	BTL-1
14.	List the steps involved in developing a decision table using tabular design notation.	Remember	BTL-1
15.	List the framework activities in the user interface analysis and design process.	Remember	BTL-1
16.	Identify the components of an architectural context diagram.	Remember	BTL-1

17.	List the steps involved in user interface analysis.		Remember	BTL-1
18.	List the steps involved in the user interface design process.		Remember	BTL-1
19.	State the facilities provided in a system for user error recovery.		Remember	BTL-1
20.	Define the concept of a user interface design pattern.		Remember	BTL-1
21.	Define data abstraction and inheritance.		Remember	BTL-1
22.	State the need for architectural mapping using a Data Flow Diagram (DFD).		Remember	BTL-1
23.	Define software architecture and design patterns.		Remember	BTL-1
24.	State the type of coupling associated with a logically cohesive module.		Remember	BTL-1
PART-B (16- MARKS)				
1.	Explain the following list of design concept (i) Abstraction and Modularity (ii) Patterns & Functional independence	(8) (8)	Evaluate	BTL-5
2.	Explain about software architecture design, with emphasize as fan in, fan-out, coupling, cohesion and factoring.	(16)	Evaluate	BTL-5
3.	Analyze your understanding on the following design models (i) Data design elements and Architectural design elements. (ii) Interface design elements and Component-level design elements.	(8) (8)	Analyze	BTL-4
4.	Demonstrate in detail about architectural design and illustrate in detail about any four architectural styles.	(16)	Apply	BTL-3
5.	Examine the transform mapping technique and analyze its steps using a suitable example.	(16)	Analyze	BTL-4
6.	Analyze the steps involved in transaction mapping and examine the transaction mapping process with a suitable example.	(16)	Analyze	BTL-4
7.	(i) Analyze the basic design principles of class-based components. (ii) Analyze the component-level design guidelines and their significance in software design.	(8) (8)	Analyze	BTL-4
8.	Describe the various coupling and cohesion methods used in software design.	(16)	Analyze	BTL-4
9.	Examine Architectural Styles. (i) Data centered Architecture and Data Flow Architecture. (ii) Call and Return Architecture and Object Oriented Architecture.	(8) (8)	Apply	BTL-3
10.	(i) Analyze on the concept of graphical design notation. (ii) Explain Tabular Design Notation.	(8) (8)	Analyze	BTL-4
11.	i) Describe about user interface analysis in detail. ii) Explain the general model of a real time system.	(8) (8)	Analyze	BTL-4
12.	Generalize on the concept of user interface design and list the characteristics of a good user interface design and Develop the design issues in interface design.	(16)	Create	BTL-6

13.	(i) Analyze about program design language in designing conventional components. (ii) Classify and explain the various architectural styles in detail.	(8) (8)	Analyze	BTL-4
14.	i) Describe how UID may be developed for a data equation system. ii) Discuss the design heuristics for effective modularity design.	(8) (8)	Analyze	BTL-4
15.	What are the good characteristics of good design? Discuss how structural partitioning can help to make software more maintainable.	(16)	Apply	BTL-3
16.	Explain the steps involved in conducting component level design When it is applied for object oriented system	(16)	Apply	BTL-3
17.	What is transform mapping? Describe the design steps of the transform mapping and transaction mapping.	(16)	Apply	BTL-3

UNIT IV- TESTING AND MAINTENANCE

Software testing fundamentals-Internal and external views of Testing-white box testing - basis path testing- control structure testing-black box testing- Regression Testing – Unit Testing – Integration Testing – Validation Testing – System Testing And Debugging –Software Implementation Techniques: Coding practices-Refactoring-Maintenance and Reengineering-BPR model-Reengineering process model-Reverse and Forward Engineering.

PART-A (2 -MARKS)

1.	Differentiate between black box testing and white box testing.	Understand	BTL-2
2.	What methods are used for breaking very long expressions and statements?	Remember	BTL-1
3.	What is the need for regression testing and system testing?	Remember	BTL-1
4.	List the levels of testing.	Remember	BTL-1
5.	What is cyclomatic complexity?	Understand	BTL-2
6.	What is a test case?	Remember	BTL-1
7.	What is a software maintenance problem?	Understand	BTL-2
8.	Define boundary value analysis.	Remember	BTL-1
9.	What is refactoring?	Understand	BTL-2
10.	What is the relationship between software testing and software reliability?	Understand	BTL-2
11.	Define reverse engineering.	Remember	BTL-1
12.	Define software reliability.	Remember	BTL-1
13.	Distinguish between alpha testing and beta testing.	Understand	BTL-2
14.	List two testing strategies that address verification. Which type of testing addresses validation?	Understand	BTL-2
15.	List the best practices for coding.	Remember	BTL-1

16.	Differentiate between verification and validation. Which type of testing addresses verification?	Understand	BTL-2
17.	What happens if software fails after acceptance testing?	Understand	BTL-2
18.	What is the difference between testing and debugging?	Understand	BTL-2
19.	What is business process reengineering?	Understand	BTL-2
20.	Who performs validation testing?	Understand	BTL-2
21.	Describe the objectives of testing. What is cyclomatic complexity?	Remember	BTL-1
22.	Give the testing principles that a software engineer must apply while performing software testing.	Understand	BTL-2
23.	Differentiate between statement coverage and branch coverage.	Understand	BTL-2
24.	What is a good test? List two principles of good test design.	Understand	BTL-2
PART-B (16- MARKS)			

1.	Discuss on i. Unit testing &Regression testing ii. Validation testing & Acceptance testing	(8) (8)	Analyze	BTL4
2.	What is Boundary value analysis? Explain the technique specifying rules and is usage with the help of an example.	(16)	Analyze	BTL4
3.	What is Equivalence class partitioning? List rules used to define valid and invalid Equivalence class. Describe the technique using example.	(16)	Apply	BTL3
4.	Elaborate path testing with an example.	(16)	Evaluate	BTL5
5.	Discuss the various Black box and white Box testing techniques. Use suitable example for your explanation.	(16)	Analyze	BTL4
6.	Describe about the various Integration & Debugging strategies followed in software development.	(16)	Analyze	BTL-4
7.	(i) Explain software implementation techniques .What is the percentage in total cost of the project? How do you expedite the implementation stage (ii) What is meant by control flow testing? Is it always falling with dataflow in case of software? Justify?	(8) (8)	Evaluate	BTL5
8.	What is black box testing? Explain the different types of black box testing strategies with example?	(16)	Create	BTL6
9.	(i) Explain the categories of debugging approaches. (ii) Why is testing important? Relate the path testing procedure in detail with sample code.	(8) (8)	Analyze	BTL4
10.	Develop BPR model to increase the efficiency of business process.	(16)	Create	BTL6
11.	Examine the software reengineering process model and analyze how refactoring contributes to software evolution.	(16)	Analyze	BTL-4
12.	What is black box testing? Explain the different types of black box testing strategies. Explain by considering suitable examples.	(16)	Analyze	BTL4
13.	(i) Highlight Forward engineering process for different types of architectures. (ii) Outline how the reverse engineering process helps the software engineer to understand the internal design structure of complex problems.	(8) (8)	Create	BTL6

14.	Describe the type's basic path testing given. (i) Flow graph notation.	(8)	Analyze	BTL4
	(ii) Independent program paths.	(8)		
15.	Summarize on Top-down Integration testing and Bottom-up integration testing.	(16)	Evaluate	BTL5
16.	(i) Illustrate in detail about Reverse engineering process.	(8)	Apply	BTL3
	(ii) Explain Forward Engineering for Client-Server Architectures.	(8)		
17.	Apply and analyze the purpose of system testing with a high level explanation on all its types.	(16)	Apply	BTL3

UNIT V-PROJECT MANAGEMENT

Software Project Management: Estimation – LOC, FP Based Estimation, Make/Buy Decision COCOMO I& II Mode – Project Scheduling – Scheduling, Earned Value Analysis Planning – Project Plan, Planning Process, RFP Risk Management – Identification, Projection - Risk Management-Risk Identification- RMMM Plan-CASE TOOLS.

PART-A (2 -MARKS)

1.	What are the decomposition techniques?	Remember	BTL-1
2.	How is the Expected Value (EV) computed for software size estimation?	Understand	BTL-2
3.	What are the different types of productivity estimation measures?	Remember	BTL-1
4.	What is a Work Breakdown Structure (WBS)?	Remember	BTL-1
5.	List any two advantages of using the COCOMO model.	Remember	BTL-1
6.	What is risk management?	Remember	BTL-1
7.	Differentiate between project risk and business risk.	Understand	BTL-2
8.	Can exhaustive testing guarantee that a program is 100% correct?	Understand	BTL-2
9.	List the activities involved in project planning.	Remember	BTL-1
10.	What is the difference between direct and indirect measures?	Understand	BTL-2
11.	How is Function Point (FP) measured?	Understand	BTL-2
12.	What is Budgeted Cost of Work Scheduled (BCWS)?	Understand	BTL-2
13.	Why is LOC not treated as a standard software metric?	Understand	BTL-2
14.	List the metrics computed during the error tracking activity.	Remember	BTL-1
15.	State the importance of scheduling activity in project management.	Understand	BTL-2
16.	Write any two differences between known risks and predictable risks.	Understand	BTL-2
17.	What is the COCOMO model used to estimate?	Understand	BTL-2
18.	How are productivity and cost associated with Function Points?	Understand	BTL-2
19.	What is Earned Value Analysis (EVA)?	Understand	BTL-2

20.	List the CASE tools used in the Design and Testing phases of SDLC.		Remember	BTL-1
21.	Define risk. What are its types? Give an example.		Remember	BTL-1
22.	What is the make-or-buy decision in software project management?		Understand	BTL-2
23.	What is Risk Mitigation, Monitoring, and Management (RMMM)?		Understand	BTL-2
24.	Differentiate between size-oriented and function-oriented metrics.		Understand	BTL-2
PART-B(16 MARKS)				
1.	Illustrate the application of decomposition methods in software cost estimation and explain the various estimation techniques with examples	(16)	Apply	BTL3
2.	(i) Apply the COCOMO I/II model to estimate the cost of a software project. (ii) Explain how different types of project plans are used in software project management.	(8) (8)	Apply	BTL3
3.	Compare and analyze the Function Point Metric Model and the three COCOMO methods for software cost estimation. Analyze the role of the Technology Complexity Factor in Function Point Analysis.	(16)	Analyze	BTL4
4.	(i) Apply the concept of risk by identifying the different types of software risks with suitable examples. (ii) Explain how the phases of risk management are applied in managing software project risks.	(8) (8)	Apply	BTL3
5.	Discuss Decision tree to support Make/buy decision.	(16)	Analyze	BTL4
6.	(i) Describe the basic principles of software project scheduling. (ii) Describe the relationship between people and effort with diagram.	(8) (8)	Analyze	BTL4
7.	(i) Point out the challenges of risk management. (ii) How to track the schedule for the project? Explain in detail.	(8) (8)	Analyze	BTL4
8.	(i) Examine the various technical metrics and measures for software? (ii) Demonstrate Software cyclomatic complexity metric with an example.	(8) (8)	Apply	BTL3
9.	Design a risk management plan for a software project by justifying the need for risk management and incorporating the major risk management activities.	(16)	Create	BTL-6
10.	Describe the following (i) Project scheduling . (ii) Project Time Line chart & Task network . (iii)	(8) (8)	Analyze	BTL4
11.	Evaluate the LOC-based and Function Point (FP)-based estimation models. Compare their features and justify which model is more suitable for software cost estimation.	(16)	Evaluate	BTL5
12.	An application has the following: 10 low external inputs, 8 high external outputs, 13 low internal logical files, 17 high external interface files, 11 average external inquiries and complexity adjustment factor of 1.10. Formulate the unadjusted and adjusted function point counts? Discuss Putnam resources allocation model. Develop the time and effort equations.	(16)	Create	BTL6

13.	Explain in detail COCOMO model for software cost estimation. Use it to estimate the effort required to build software for a simple ATM that produces 12 screens,10 reports and has 80 software components. Assume average complexity and average developer maturity. Use application composition model with object points.	(16)	Analyze	BTL4
14.	Describe in detail about the following scheduling (i) Timeline charts. (ii) Tracking the schedule and Tracking progress for an OO project.	(8) (8)	Analyze	BTL4
15.	(i) Discuss about risk management in a software development life cycle. (ii) Discuss on the concept of RMMM.	(8) (8)	Analyze	BTL4
16.	Demonstrate on the following list given below (i) Function Point estimation. (ii) LOC based estimation.	(8) (8)	Apply	BTL3
17.	(i) Explain in detail about risk identification. (ii) Analyze on the concept of risk Projection.	(8) (8)	Analyze	BTL4