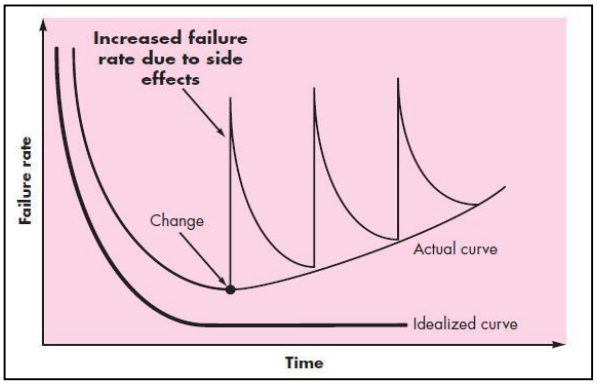
Winter – 19 EXAMINATION

Subject Name: Software Engineering **Model Answer** **Subject Code: 22413**

Important Instructions to examiners:

1) The answers should be examined by key words and not as word-to-word as given in the model answer scheme.
2) The model answer and the answer written by candidate may vary but the examiner may try to assess the understanding level of the candidate.
3) The language errors such as grammatical, spelling errors should not be given more Importance (Not applicable for subject English and Communication Skills.
4) While assessing figures, examiner may give credit for principal components indicated in the figure. The figures drawn by candidate and model answer may vary. The examiner may give credit for any equivalent figure drawn.
5) Credits may be given step wise for numerical problems. In some cases, the assumed constant values may vary and there may be some difference in the candidate's answers and model answer.
6) In case of some questions credit may be given by judgement on part of examiner of relevant answer based on candidate's understanding.
7) For programming language papers, credit may be given to any other program based on equivalent concept.

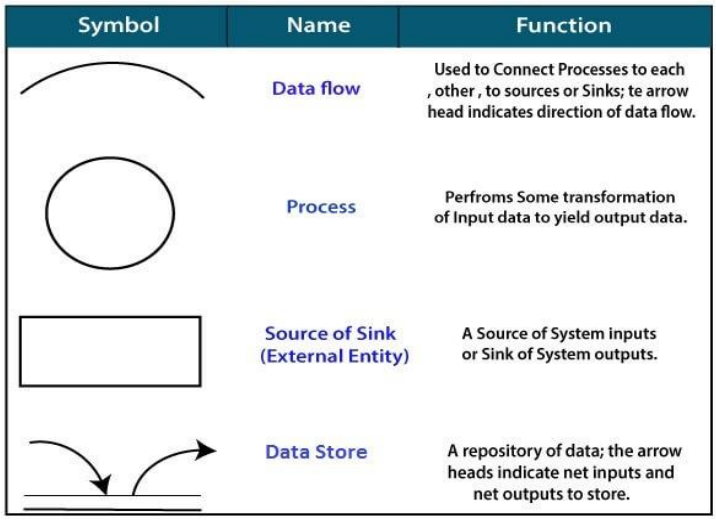
Q. No.	Sub Q. N.	Answer	Marking Scheme
1.		**Attempt any Five of the following:**	**10M**
	a	**Define software. Draw the failure curve for software.**	**2M**
	Ans	Definition of Software Software is: 1. Instructions (computer programs) that when executed provide desired features, function, and performance; 2. Data structures that enable the programs to adequately manipulate information, and 3. Descriptive information (documents) in both hard copy and virtual forms that describes the operation and use of the programs.  	Correct definition 1M and diagram 1M

	b	**State two characteristics of Software.**	**2M**
	Ans	**Characteristics of software :** • Software is developed or engineered; it is not manufactured in the classical sense. • Software doesn't "wear out." But it does deteriorate! • Although the industry is moving toward component-based construction, most software continues to be custom built.	Any two correct Characteristics - 1M each
	c	**Define software requirement specification**	**2M**
	Ans	**Concept:** A software requirements specification (SRS) is a document that is created when a detailed description of all aspects of the software to be built that must be specified before the project is to commence. It is a primary document for development of software. It is written by Business Analysts who interact with client and gather the requirements to build the software.	Correct definition -2M
	d	**Define proactive and reactive risk strategy.**	**2M**
	Ans	**Reactive risk strategies** • Reactive risk strategy follows that the risks have to be tackled at the time of their occurrence. • No precautions are to be taken as per this strategy. • They are meant for risks with relatively smaller impact. • More commonly, the software team does nothing about risks until something goes wrong. • Then, the team flies into action in an attempt to correct the problem rapidly. This is often called a fire-fighting mode. • **Proactive risk strategies** • It follows that the risks have to be identified before start of the project. • They have to be analysed by assessing their probability of occurrence, their impact after occurrence, and steps to be followed for its precaution.	Correct definition -1M each
	e	**Name two cost estimation approaches.**	**2M**
	Ans	• Heuristic Estimation Approach • Analytical Estimation Approach • Empirical Estimation Approach	Any two techniques-1M each
	f	**Define software quality.**	**2M**
	Ans	1.Quality means that a product satisfies the demands of its specifications 2. It also means achieving a high level of customer satisfaction with the product 3. In software systems this is difficult • Customer quality requirements(e.g. efficiency or reliability) often conflict with developer quality requirements (e.g. maintainability or reusability)	Correct Definition-2M

		• Software specifications are often incomplete, inconsistent, or ambiguous	
	g	**Name four software quality assurance activities.**	**2M**
	Ans	These activities are performed (or facilitated) by an independent SQA group that: i. Prepares an SQA plan for a project. ii. Participates in the development of the project's software process description. iii. Reviews software engineering activities to verify compliance with the defined software process. iv. Audits designated software work products to verify compliance with those defined as part of the software process. v. Ensures that deviations in software work and work products are documented and handled according to a documented procedure. vi. Records any noncompliance and reports to senior management.	Any 4 activity name-1/2M each
2.		**Attempt any Three of the following:**	**12M**
	a	**State and explain with examples four categories of software.**	**4M**
	Ans	**Types / Categories of Software** **1. System Software** 1. System software is a collection of programs written to service other programs. 2. Few examples of system software are compilers, editors, and file management utilities, process complex, but determinate, information structures. 3. Other systems applications are operating system components, drivers, and telecommunications. Example : DOS, WINDOWS **2. Real-time Software** ***(*Question: Explain the features of real world software. – 3 Marks*)*** 1. Software that monitors or analyses or controls real-world events as they occur is called real time. 2. Elements of real-time software include a data gathering component that collects and formats information from an external environment, an analysis component that transforms information as required by the application. 3. A control/output component that responds to the external environment and a monitoring component that coordinates all other components so that real-time response can be maintained. Example : airline reservation system, railway reservation system **3. Business Software** 1. Business information processing is the largest single software application area. Discrete "systems".	Any 4 types explanation with example-4M

		2. For example: payroll, accounts receivable/payable, inventory have evolved into management information system (MIS) software that accesses one or more large databases containing business information. 3. Applications in this area restructure existing data in a way that facilitates business operations or management decision making. 4. In addition to conventional data processing application, business software applications also encompass interactive computing. Example : Tally **4. Engineering and Scientific Software** 1. Engineering and scientific software have been characterized by "number crunching" algorithms. 2. Applications range from astronomy to volcanology, from automotive stress analysis to space shuttle orbital dynamics, and from molecular biology to automated manufacturing. 3. However, modern applications within the engineering/scientific area are moving away from conventional numerical algorithms. 4. Computer-aided design, system simulation, and other interactive applications have begun to take on real-time and even system software characteristics. Example : CAD / CAM software **5. Embedded Software** 1. Intelligent products have become commonplace in nearly every consumer and industrial market. 2. Embedded software resides in read-only memory and is used to control products and systems for the consumer and industrial markets. 3. Embedded software can perform very limited and esoteric functions, for example: keypad control for a microwave oven. 4. To provide significant function and control capability, for example: digital functions in an automobile such as fuel control, dashboard displays, and braking systems. Example : Microwave, Washing machine software **6. Personal Computer Software** 1. The personal computer software market has burgeoned over the past two decades. 2.Word processing, spread sheets, computer graphics, multimedia, entertainment, database management, personal and business fi applications, external network, and database access are only a few of hundreds of applications. Example: Microsoft word, Excel.	
	b	**Explain the notations used for preparing a Data Flow diagram.**	**4M**
	Ans	**Circle:** A circle (bubble) shows a process that transforms data inputs into data outputs. **Data Flow:** A curved line shows the flow of data into or out of a process or data store.	Correct symbols with explanation -1M each

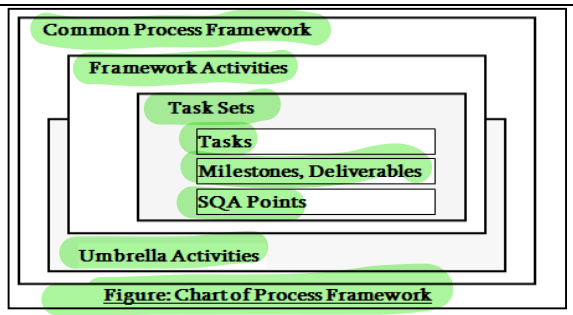
Data Store: A set of parallel lines shows a place for the collection of data items. A data store indicates that the data is stored which can be used at a later stage or by the other processes in a different order. The data store can have an element or group of elements.
Source or Sink: Source or Sink is an external entity and acts as a source of system inputs or sink of system outputs.



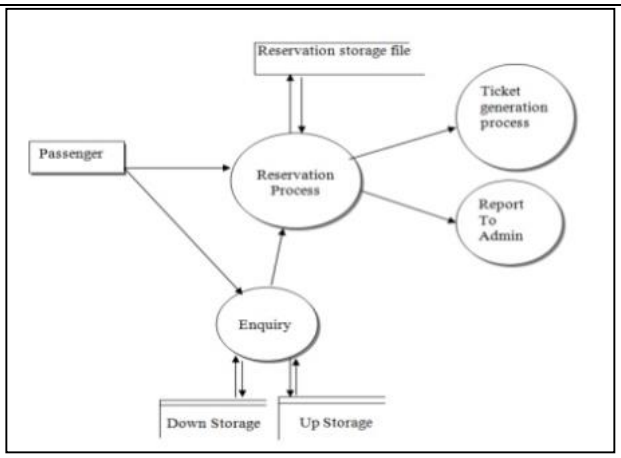
Symbols for Data Flow Diagrams

	c	**Describe 4 P's of management spectrum giving their significance.**	**4M**
	Ans	**The Management Spectrum – 4 Ps and their Significance** Effective software project management focuses on these items (in this order) Deals with the cultivation of motivated, highly skilled people **1. The people** i. Consists of the stakeholders, the team leaders, and the software team **2. The product** i. Product objectives and scope should be established before a project can be planned. **3. The process** i. The software process provides the framework from which a comprehensive plan for software development can be established. **4. The project** i. Planning and controlling a software project is done for one primary reason…it is the only known way to manage complexity ii. In a 1998 survey, 26% of software projects failed outright, 46% experienced cost and schedule overruns.	Description of each P's-1M each
	d	**Explain four basic principles of software project scheduling.**	

	Ans	Basic principles software project scheduling: **Compartmentalization:** The project must be compartmentalized into a number of manageable activities and tasks. To accomplish compartmentalization, both the product and the process are Decomposed. **Interdependency:** The interdependency of each compartmentalized activity or task must be determined. Some tasks must occur in sequence while others can occur in parallel. Some activities cannot commence until the work product produced by another is available. Other activities can occur independently. **Time allocation:** Each task to be scheduled must be allocated some number of work units (e.g., person-days of effort). In addition, each task must be assigned a start date and a completion date that are a function of the interdependencies and whether work will be conducted on a fulltime or part-time basis. **Effort validation:** Every project has a defined number of staff members. As time allocation occurs, the project manager must ensure that no more than the allocated number of people has been scheduled at any given time. **Defined responsibilitie**s: Every task that is scheduled should be assigned to a specific team member. Defined outcomes: Every task that is scheduled should have a defined outcome. **Defined milestones**: Every task or group of tasks should be associated with a project milestone. Program evaluation and review technique (PERT) and critical path method (CPM) are two project scheduling Methods that can be applied to software development. **Defined outcomes** – Every task that is scheduled should have a defined outcome for software projects such as a work product or part of a work product – Work products are often combined in deliverables	Any four correct principles -1M each
3.		**Attempt any Three of the following:**	**12M**
	a	**Explain Process framework with a suitable diagram.**	**4M**
	Ans	A process framework establishes the foundation for a complete software process by identifying a small number of framework activities that are applicable to all software projects; In addition, the process framework encompasses a set of umbrella activities that are applicable across the entire software process.	Description 2M Diagram 2 M

		 Figure: Chart of Process Framework Basic framework activities: **1. Communication**: This framework activity involves heavy communication & collaboration with the customer (and the stakeholders) and encompasses requirements gathering and other related activities. **2. Planning**: This activity establishes a plan for the software engineering work that follows. It describes the technical tasks to be conducted; the risks are analyzed. Project tracking should be done. Deadline is fixed. **3. Modeling**: This activity encompasses the creation of models that allow the developer & the customer to better understand software requirements & the design that will achieve those requirements. **4. Construction**: This activity combines code generation and the testing that is required uncovering errors in the code. **5. Deployment**: The software is delivered to the customer who evaluates the delivered product and provides feedback based on the evaluation.	
	b	**Describe four principles of good planning.**	**4M**
	Ans	**Principle 1. Understand the scope of the project.** It's impossible to use a road map if you don't know where you're going. Scope provides the software team with a destination. **Principle 2. Involve stakeholders in the planning activity.** Stakeholders define priorities and establish project constraints. To accommodate these realities, software engineers must often negotiate order of delivery, time lines, and other project-related issues. **Principle 3. Recognize that planning is iterative.** A project plan is never engraved in stone. As work begins, it is very likely that things will change. As a consequence, the plan must be adjusted to accommodate these changes. In addition, iterative, incremental process models dictate re-planning after the delivery of each software increment based on feedback received from users. **Principle 4. Estimate based on what you know.** The intent of estimation is to provide an indication of effort, cost, and task duration, based on the team's current understanding of the work to be done. If information is vague or unreliable, estimates will be equally unreliable. **Principle 5.Consider risk as you defines the plan.** If you have identified risks that have high impact and high probability, contingency planning is necessary. In addition, the project plan (including the schedule) should be	Any 4 Principles; 1 M each

		adjusted to accommodate the likelihood that one or more of these risks will occur. **Principle 6. Be realistic.** People don't work 100 percent of every day. Noise always enters into any human communication. Omissions and ambiguity are facts of life. Change will occur. Even the best software engineers make mistakes. These and other realities should be considered as a project plan is established. **Principle 7.Adjust granularity as you defines the plan.** Granularity refers to the level of detail that is introduced as a project plan is developed. A high-granularity plan provides significant work task detail that is planned over relatively short time increments (so that tracking and control occur frequently). A low-granularity plan provides broader work tasks that are planned over longer time periods. In general, granularity moves from high to low as the project time line moves away from the current date. Over the next few weeks or months, the project can be planned in significant detail. Activities that won't occur for many months do not require high granularity (too much can change). **Principle 8. Define how you intend to ensure quality.** The plan should identify how the software team intends to ensure quality. If technical reviews are to be conducted, they should be scheduled. If pair programming is to be used during construction, it should be explicitly defined within the plan. **Principle 9. Describe how you intend to accommodate change.** Even the best planning can be obviated by uncontrolled change. You should identify how changes are to be accommodated as software engineering work proceeds. For example, can the customer request a change at any time? If a change is requested, is the team obliged to implement it immediately? How is the impact and cost of the change assessed? **Principle 10.Track the plan frequently and make adjustments as required.** Software projects fall behind schedule one day at a time. Therefore, it makes sense to track progress on a daily basis, looking for problem areas and situations in which scheduled work does not conform to actual work conducted. When slippage is encountered, the plan is adjusted accordingly.	
	c	**Draw and explain Level 1 DFD for railway reservation system.**	**4M**
	Ans		Diagram 2 M Description 2



The passenger can initiate either Reservation process or Enquiry process; If a user opts for Reservation process then the system shall proceed with ticket generation process and same needs to be notified to the Admin. If user opts for enquiry module then appropriate request shall be entertain and result to be displayed to the user.

d **With an example, explain Line of Code (LOC) based estimation.** **4M**

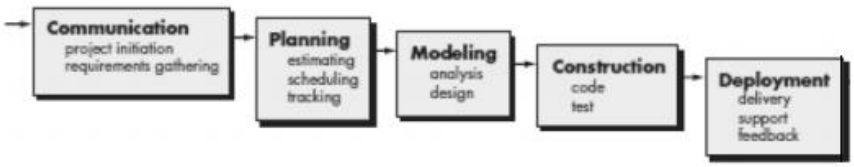
Ans

LOC-Based Estimation: As an example of LOC and FP problem-based estimation techniques, let us consider a software package to be developed for a computer-aided design application for mechanical components.
A review of the System Specification indicates that the software is to execute on an engineering workstation and must interface with various computer graphics peripherals including a mouse, digitizer, high resolution color display and laser printer.
Using the System Specification as a guide, a preliminary statement of software scope can be developed:

Example:

Function	**Estimated LOC**
User interface and control facilities (UICF)	2,300
Two-dimensional geometric analysis (2DGA)	5,300
Three-dimensional geometric analysis (3DGA)	6,800
Database management (DBM)	3,350
Computer graphics display facilities (CGDF)	4,950
Peripheral control function (PCF)	2,100
Design analysis modules (DAM)	8,400

Description 2M
Example 2M

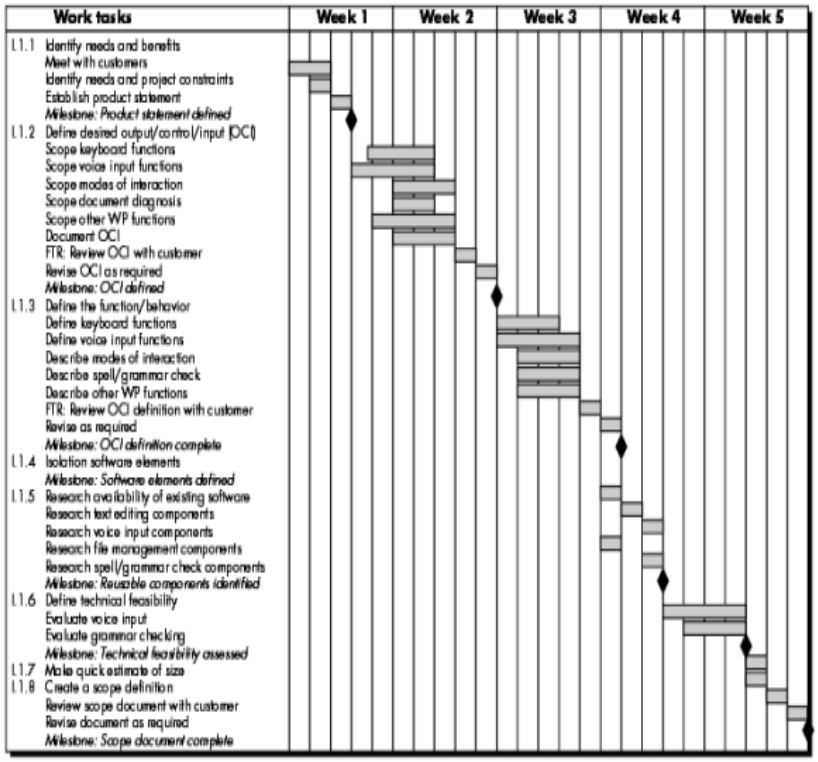
		Estimated lines of code | 33,200	
4.		**Attempt any Three of the following:**	**12M**
	a	**Explain waterfall process model. State its advantages and disadvantages.**	**4M**
	Ans	 The waterfall model is a traditional method, sometimes called the classic life cycle. This is one of the initial models. As the figure implies stages are cascaded and shall be developed one after the other. It suggests a systematic, sequential approach to software development that begins with customer specification of requirements and progresses through, communication, planning, modeling construction and deployment. In other words, one stage should be completed before the other begins. Hence, when all the requirements are elicited by the customer, analyzed for completeness and consistency, documented as per requirements, the development and design activities commence. One of the main needs of this model is the user 's explicit prescription of complete requirements at the start of development. For developers it is useful to layout what they need to do at the initial stages. Its simplicity makes it easy to explain to customers who may not be aware of software development process. It makes explicit with intermediate products to begin at every stage of development. One of the biggest limitations is it does not reflect the way code is really developed. Problem is well understood but software is developed with great deal of iteration. Often this is a solution to a problem which was not solved earlier and hence software developers shall have extensive experience to develop such application; as neither the user nor the developers are aware of the key factors affecting the desired outcome and the time needed. Hence at times the software development process may remain uncontrolled. Today software work is fast paced and subject to a never-ending stream of changes in features, functions and information content. Waterfall model is inappropriate for such work. This model is useful in situation where the requirements are fixed and work proceeds to completion in a linear manner. **Advantages of waterfall model:** 1. This model is simple and easy to understand and use. 2. It is easy to manage due to the rigidity of the model – each phase has specific deliverables and a review process.	Description 2M Any 2 advantage 1M Any 2 Disadvantages 2M

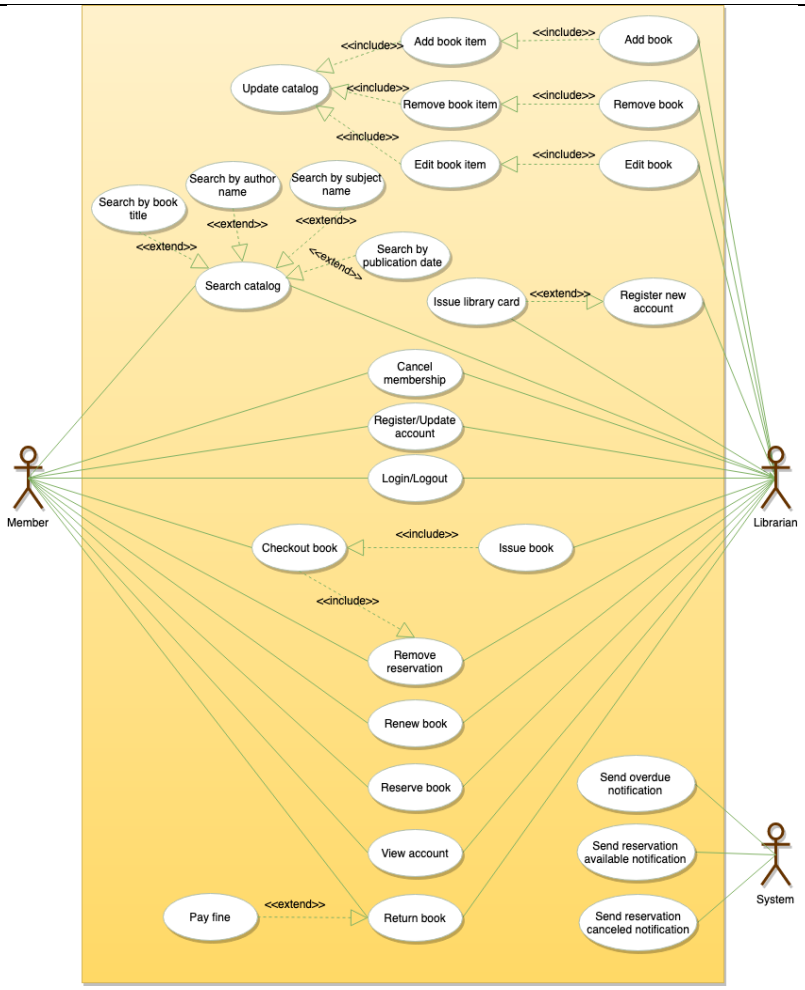
		3. In this model phases are processed and completed one at a time. Phases do not overlap. 4. Waterfall model works well for smaller projects where requirements are very well understood. **Disadvantages of waterfall model:** 1. Once an application is in the testing stage, it is very difficult to go back and change something that was not well-thought out in the concept stage. 2. No working software is produced until late during the life cycle. 3. High amounts of risk and uncertainty. 4. Not a good model for complex and object-oriented projects. 5. Poor model for long and ongoing projects. 6. Not suitable for the projects where requirements are at a moderate to high risk of changing.	
	b	**Enlist core principles of software engineering practice.**	**4M**
	Ans	1. Reason it all exists. Provide value to the user 2.Keep it simple stupid 3.Maintain the vision 4. What you reproduce, someone else will have to consume. (implement knowing someone else will have to understand what you are doing) 5.Be open to the future 6. Plan ahead for reuse Plan ahead for reuse Think!	List of all 7 core principles 4M
	c	**Describe RMMM Strategy.**	**4M**
	Ans	Risk mitigation, monitoring, and management (RMMM) plan. A risk management strategy can be included in the software project plan or the risk management steps can be organized into a separate Risk Mitigation, Monitoring and Management Plan. The RMMM plan documents all work performed as part of risk analysis and is used by the project manager as part of the overall project plan. Once RMMM has been documented and the project has begun, risk mitigation and monitoring steps commence. Risk mitigation is a problem avoidance activity. Risk monitoring is a project tracking activity with three primary objectives: (1) To assess whether predicted risks do, in fact, occur; (2) To ensure that risk aversion steps defined for the risk are being properly applied; and (3) To collect information that can be used for future risk analysis. In many cases, the problems that occur during a project can be traced to more than one risk. Another job of risk monitoring is to attempt to allocate origin (what risk(s) caused which problems throughout the project). An effective strategy must consider three issues: • Risk avoidance • Risk monitoring • Risk management and contingency planning.	Description 4M any relevant description shall be considered

		If a software team adopts a proactive approach to risk, avoidance is always the best strategy. This is achieved by developing a plan for risk mitigation. To mitigate this risk, project management must develop a strategy for reducing turnover. Among the possible steps to be taken are • Meet with current staff to determine causes for turnover (e.g., poor working conditions, low pay, and competitive job market). • Mitigate those causes that are under our control before the project starts. • Once the project commences, assume turnover will occur and develop techniques to ensure continuity when people leave. • Organize project teams so that information about each development activity is widely dispersed. • Define documentation standards and establish mechanisms to be sure that documents are developed in a timely manner. • Conduct peer reviews of all work (so that more than one person is "up to speed). • Assign a backup staff member for every critical technologist. As the project proceeds, risk monitoring activities commence. The project manager monitors factors that may provide an indication of whether the risk is becoming more or less likely. In the case of high staff turnover, the following factors can be monitored: • General attitude of team members based on project pressures. • The degree to which the team has jelled. • Interpersonal relationships among team members. • Potential problems with compensation and benefits. • The availability of jobs within the company and outside it. In addition to monitoring these factors, the project manager should monitor the effectiveness of risk mitigation steps. RMMM steps incur additional project cost. Part of risk management, therefore, is to evaluate when the benefits accrued by the RMMM steps are outweighed by the costs associated with implementing them. In essence, the project planner performs a classic cost/benefit analysis.	
	d	**Describe the Analytical method of project cost estimation with example.**	**4M**
	Ans	Analytical estimation techniques derive the required results starting with basic assumptions regarding the project. Thus, unlike empirical and heuristic techniques, analytical techniques do have scientific basis. **Halstead's software science is an example of an analytical technique.** Halstead's software science can be used to derive some interesting results starting with a few simple assumptions. Halstead's software science is especially useful for estimating software maintenance efforts. In fact, it outperforms both empirical and heuristic techniques when used for predicting software maintenance efforts. Halstead's Software Science – An Analytical Technique Halstead's software science is an analytical technique to measure size, development	Description 2M Example 2M

		effort, and development cost of software products. Halstead used a few primitive program parameters to develop the expressions for overall program length, potential minimum value, actual volume, effort, and development time. For a given program, let:
		• η1 be the number of unique operators used in the program,
		• η2 be the number of unique operands used in the program,
		• N1 be the total number of operators used in the program,
		• N2 be the total number of operands used in the program.
		Example: Let us consider the following C program:
		main( )
		{ int a, b, c, avg;
		scanf("%d %d %d", &a, &b, &c);
		avg = (a+b+c)/3;
		printf("avg = %d", avg);
		} The unique operators are: main, (), {}, int, scanf, &, ", ", =, +, /, printf
		The unique operands are: a, b, c, &a, &b, &c, a+b+c, avg, 3, "%d %d %d", "avg = %d"
		Therefore, η1 = 12, η2 = 11
		Estimated Length = (12*log12 + 11*log11)
		= (12*3.58 + 11*3.45)
		= (43+38) = 81
		Volume = Length*log(23)
		= 81*4.52
		= 366	

	e	**Explain GANTT chart and its application for project tracking with an example.**	**4M**
	Ans	When creating software project schedule, we begin with a set of tasks. If automated tools are used, the work breakdown is input as a task network or task outline. Effort, duration and start date are then input for each task, In addition, tasks may be assigned to specific individuals. As a consequence of this input, a time-line chart, also called a Gantt chart is generated. A time-line chart can be developed for the entire project. The figure below depicts a part of a software project schedule that emphasizes scoping task for a word-processing (WP) software product. All project tasks are listed in the left-hand column. The horizontal bars indicate the duration of each task. When multiple bars occur at the same time on the calendar, task concurrency is implied. The diamond indicates milestones. Once the information necessary for the generation of a time-line chart has been input, the majority of software project scheduling tools produce project tables – a tabular listing of all project tasks, their planned and actual start and end dates, and a variety of related information. Used in conjunction with the time-line chart, project tables enable to track progress.	Description and Example 3M Application1M

		Time-Line chart - Micro-level Scheduling  Application of Gantt Chart • The sheer simplicity and ease-of-access of all relevant information make Gantt charts an ideal choice for teams to use them for organizing their schedules. Due to this, Gantt charts are widely used in project management, IT and development teams. • Apart from them, marketing, engineering, product launch, manufacturing teams can also use Gantt charts to get an overview of how things are rolling on the work front.	
5.		**Attempt any Two of the following:**	**12M**
	a	**Sketch a use case diagram for library management system with minimum four use cases and two actors**.	**6M**
	Ans		Correct/relevant any four use cases -6M

			
	b	**Explain the concept of black box testing and white box testing.**	**6M**
	Ans	**Black Box Testing:** • It is a way of software testing in which the internal structure or the program or the code is hidden and nothing is known about it. • It also known as data-driven, box testing, data-, and functional testing. • This type of testing is ideal for higher levels of testing like System Testing, Acceptance testing. • It is mostly done by software testers. • No knowledge of implementation is needed. • It is functional test of the software. • Testing can start after preparing requirement specification document.	Black box testing explanation -3M and white box testing explanation- 3M

- Techniques used:
  - **Equivalence partitioning**: Equivalence partitioning divides input values into valid and invalid partitions and selecting corresponding values from each partition of the test data.
  - **Boundary value analysis:** checks boundaries for input values.
- **Advantages of Black Box Testing**
  - Efficient when used on large systems.
  - Since the tester and developer are independent of each other, testing is balanced and unprejudiced.
  - Tester can be non-technical.
  - There is no need for the tester to have detailed functional knowledge of system.
  - Tests will be done from an end user's point of view, because the end user should accept the system. (This testing technique is sometimes also called Acceptance testing.)
  - Testing helps to identify vagueness and contradictions in functional specifications.
  - Test cases can be designed as soon as the functional specifications are complete.
- **Disadvantages of Black Box Testing**
  - Test cases are challenging to design without having clear functional specifications.
  - It is difficult to identify tricky inputs if the test cases are not developed based on specifications.
  - It is difficult to identify all possible inputs in limited testing time. As a result, writing test cases may be slow and difficult.
  - There are chances of having unidentified paths during the testing process.
  - There is a high probability of repeating tests already performed by the programmer.

White Box Testing:

- It is a way of testing the software in which the tester has knowledge about the internal structure r the code or the program of the software.
- It is also called structural testing, clear box testing, code-based testing, or glass box testing.

		• Testing is best suited for a lower level of testing like Unit Testing, Integration testing.
		• It is mostly done by software developers.
		• Knowledge of implementation is required.
		• It is structural test of the software.
		• Testing can start after preparing for Detail design document.
		• Techniques Used:
		  ○ Statement Coverage, Branch coverage, and Path coverage are White Box testing technique.
		  ○ **Statement Coverage** validates whether every line of the code is executed at least once.
		  ○ **Branch coverage** validates whether each branch is executed at least once.
		  ○ **Path coverage** method tests all the paths of the program.
		• **Advantages of White Box Testing**
		  • Code optimization by finding hidden errors.
		  • White box tests cases can be easily automated.
		  • Testing is more thorough as all code paths are usually covered.
		  • Testing can start early in SDLC even if GUI is not available.
		• **Disadvantages of White Box Testing**
		  White box testing can be quite complex and expensive.
		  • Developers who usually execute white box test cases detest it. The white box testing by developers is not detailed can lead to production errors.
		  • White box testing requires professional resources, with a detailed understanding of programming and implementation.
		  • White-box testing is time-consuming, bigger programming applications take the time to test fully.	

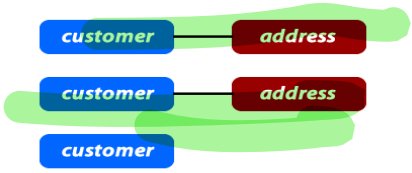
	c	**Calculate using COCOMO model** **i)Effort** **ii)Project duration** **iii)Average staff size** **If estimated size of project is 200 KLOC using organic mode.**	**6M**

	Ans	Given data: size=200 KLOC mode= organic 1. Effort: $E = a (KLOC)^{b}$ For organic a=2.4 and b= 1.05 $E= 2.4 (200)^{1.05}$ = 626 staff members 2. Project duration: $TDEV= c (E)^{d}$ Where TDEV= time for development c and d are constant to be determined E = effort For organic mode, c= 2.5 and d= 0.38 $TDEV= 2.5 (626)^{0.38}$ = 29 months 3. Average staff size: SS = E/TDEV SS = 626/29 = 22 staffs	Correct Answer for each point asked -6M
6.		**Attempt any Two of the following:**	**12M**
	a	**Define data objects, attributes, relationship, and cardinality, with example of each.**	**6M**
	Ans	**Data Object:** A data object is an entity/object in the real world with an independent existence that can be differentiated from other objects. Example: An entity might be ○ An object with physical existence (e.g., a lecturer, a student, a car) ○ An object with conceptual existence (e.g., a course, a job, a position)	Definition of each one-4M and example of each-2M

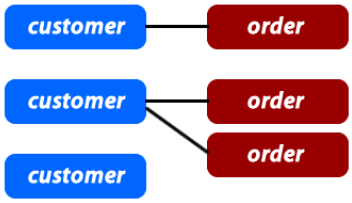
Attributes: Each data object/ entity is described by a set of attributes (e.g., Employee = (Name, Address, Birthdate (Age), Salary).

Each attribute has a name, and is associated with an entity and a domain of legal values.
Example: Employee = (Name, Address, Birthdate (Age), Salary).

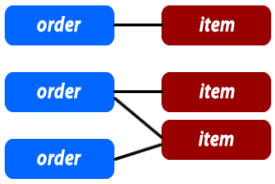
Relationship: A relationship identifies names and defines an association between two entity types **One**-to-one relationship: Example: We have a relationship between the Customers table and the Addresses table. If each address can belong to only one customer, this relationship is "One to One".



One –to – many relationship:
Example:
Each customer may have zero, one or multiple orders. But an order can belong to only one customer.



Many- to – many Relationship:
Example: In some cases, you may need multiple instances on both sides of the relationship. For example, each order can contain multiple items. And each item can also be in multiple orders.

		 **Cardinality:** In the case of Data Modeling, **Cardinality** defines the number of attributes in one entity set, which can be associated with the number of attributes of other set via relationship set. Example: **One-to-one, One-to-many, Many-to-one, Many-to-many.**	
	b	**Compare CMMI and ISO for software w.r.to** **i)scope** **ii)Approach** **Iii) Implementation.**	**6M**
	Ans	**Difference between CMMI and ISO based on** **SCOPE:** CMMI is rigid and extends only to businesses developing software intensive systems. ISO is flexible and applicable to all manufacturing industries. CMMI focuses on engineering and project management processes whereas ISO's focus is generic in nature. CMMI mandates generic and specific practices and businesses have a choice of selecting the model relevant to their business needs from 22 developed process areas. ISO requirements are same for all companies, industries, and disciplines. **APPROACH:**CMMI requires ingraining processes into business needs so that such processes become part of corporate culture and do not break down under the pressure of deadlines. ISO specifies to conformance and remains oblivious as to whether such conformance is of strategic business value or not.CMMI approaches risk management as an organized and technical discipline by identifying risk factors, quantifying such risk factors, and tracking them throughout the project life cycle. ISO was until recently neutral on risk management. ISO 31000:2009 now provides generic guidelines for the design, implementation, and maintenance of risk management processes throughout an organization.	Difference based on Scope-2M Approach-2M and Implementation 2M

		Although CMMI focuses on linkage of processes to business goals, customer satisfaction is not a factor in the ranking whereas customer satisfaction is an important part of ISO requirements. **IMPLEMENTATION:** Neither CMMI nor ISO requires the establishment of new processes. CMMI compares the existing processes to industry best practices whereas ISO requires adjustment of existing processes to confirm to the specific ISO requirements. In practice, some organizations tend to rely on extensive documentation while implementing both CMMI and ISO. Most organizations tend to constitute in-house teams, or rely on external auditors to see through the implementation process.	
	c	**Explain six function of requirement engineering process.**	**6M**
	Ans	**Requirement Engineering:** The broad spectrum of tasks and techniques that lead to an understanding of requirements is called requirements engineering. It starts during the communication activity and continues into the modeling activity. Requirements engineering provides the appropriate mechanism for understanding what the customer wants by analyzing need, assessing feasibility negotiating a reasonable solution, specifying the solution ambiguously, validating the specification, and managing the requirements as they are transformed into an operational system. It encompasses seven distinct tasks: inception, elicitation, elaboration, negotiation, specification, validation, and management. **Inception:** The question why you want to do this will be answered and analyses to identify business need, a potential new market with breadth and depth and services to be provided. The above points establish a basic understanding of the problem, the people who want a solution, the nature of the solution that is desired to understand the scope of the project. **Elicitation:** This answers for what are things need to do by asking the customer, the users, and others what the objectives for the system or product are, what is to be accomplished, how the system or product fits into the needs of the business, and finally, how the system or product is to be used on a day-to-day basis **Elaboration:** The information obtained from the customer during inception and elicitation is expanded and refined during elaboration. This	Correct/relevant explanation for each function- 1M

task focuses on developing a refined requirements model that identifies requirements for three domains, information, functional and behavioral domain. It

- Describe how the end user (and other actors) will interact with the system.
- Business domain entities that is visible to the end user.
- The attributes of each analysis class are defined, and the services that are required by each class are identified.
- The relationships and collaboration between classes are identified, and a variety of supplementary diagrams are produced.

Negotiation: It answers for is it actually required? Through which Customers, users, and other stakeholders are asked to rank requirements and prioritized the same. Using an iterative approach that prioritizes requirements, assesses their cost and risk, and addresses internal conflicts, requirements are eliminated, combined, and/or modified so that each party achieves some measure of satisfaction.

Specification: A specification can be a written document, a set of graphical models, a formal mathematical model, a collection of usage scenarios, a prototype, or any combination of these to present gathered requirements. The formality and format of a specification varies with the size and the complexity of the software to be built.

Validation: As a part of this task documented software requirement specification will be examining by conducting technical reviews in order to examine errors in content or interpretation, areas where clarification may be required, missing information, inconsistencies (a major problem when large products or systems are engineered), conflicting requirements, or unrealistic (unachievable) requirements.

Requirements management: Requirements management is a set of activities that help the project team identify, control, and track requirements and changes to requirements at any time as the project proceeds.