CS8494 - SOFTWARE ENGINEERING - 2 MARKS

UNIT – 1

PART –A

S.NO	QUESTIONS	CO	BLOOM'S LEVEL
1	**Write down the generic process framework that is applicable to any software project / relationship between work product, task, activity and system <u>NOV/DEC-10,NOV/DEC2016, NOV/DEC 2017</u>** Common process frame work - Process frame work activities - Umbrella activities - Frame work activities - Task sets	C215.1	**BTL6**
2	**List the goals of software engineering? <u>APR/MAY-11</u>** Satisfy user requirements , High reliability , Low maintenance cost , Delivery on time , Low production cost , High performance , Ease of reuse.	C215.1	**BTL6**
3	**What is the difference between verification and validation? <u>NOV/DEC-10 , APR/MAY-11 , NOV/DEC-11, MAY/JUN-13</u>** • Verification refers to the set of activities that ensure that software correctly implements a specific function. Verification: "Are we building the product right?"	C215.1	**BTL5**

	- Validation refers to a different set of activities that ensure that the software that has been built is traceable to customer requirements. Validation: "Are we building the right product?"		
4	**For the scenario described below, which life cycle model would you choose? Give the reason why you would choose this model. <u>NOV/DEC-11 ,</u>** You are interacting with the MIS department of a very large oil company with multiple departments. They have a complex regency system. Migrating the data from this legacy system is not an easy task and would take a considerable time. The oil company is very particular about processes, acceptance criteria and legal contracts. Spiral model Proactive problem prevention. Each iteration has a risk analysis, sector that evaluates. Alternatives for proactive problem avoidance.	C215.1	**BTL6**
5	**Give two reasons why system engineers must understand the environment of a system? <u>APR/MAY-12</u>** 1.The reason for the existence of a system is to make some changes in its environment. 2.The functioning of a system can be very difficult to predict.	C215.1	**BTL6**
6	**What are the two types of software products? <u>APR/MAY-12</u>** 1. Generic products: these are stand-alone systems that are produced by a development Organization and sold in the open market to any customer who wants to buy it. 2. Customized products: these are systems that are commissioned by a specific customer and developed specially by some contractor to meet a special need.	C215.1	**BTL5**
7	**What is the advantage of adhering to life cycle models for software? <u>NOV/DEC-12</u>** It helps to produce good quality software products without time and cost over runs.It encourages the development of software in a systematic & disciplined	C215.1	**BTL6**

	manner.		
8	**Is it always possible to realize win-win spiral model for software? Justify. NOV/DEC-12** ○ Must identify stake holder and their win condition ○ Developing buy-in to the model is important than the model itself ○ Eliminating the clashes between customers is important.	C215.1	**BTL6**
9	**What is software process? List its activities. MAY/JUN-13** Software process is defined as the structured set of activities that are required to develop the software system. Activities – Specification, design & implementation, validation & evolution.	C215.1	**BTL6**
10	**What are the various categories of software?** • System software • Application software • Engineering/Scientific software • Embedded software • Web Applications • Artificial Intelligence software	C215.1	**BTL5**
11	**What are the umbrella activities of a software process? APR/MAY 2015** • Software project tracking and control. • Risk management. • Software Quality Assurance. • Formal Technical Reviews. • Software Configuration Management. • Work product preparation and production. • Reusability management. • Measurement	C215.1	**BTL6**
12	**What are the merits of incremental model?** i. The incremental model can be adopted when tere are less number of people involved in the project.	C215.1	**BTL6**

	ii. Technical risks can be managed with each increment. iii. For a very small time span,at least core product can be delivered to the customer.		
13	**List the task regions in the Spiral model**. • Customer communication – In this region it is suggested to establish customer communication. • Planning – All planning activities are carried out in order to define resources timeline and otherproject related activities. • Risk analysis – The tasks required to calculate technical and management risks. • Engineering – In this the task region,tasks required to build one or more representations of applications are carried out. • Construct and release – All the necessary tasks required to construct,test,install the applications are conducted. ¾_Customer evaluation – Customer‟ s feedback is obtained and based on the customer evaluation required tasks are performed and implemented at installation stage.	C215.1	**BTL5**
14	**Characteristics of software contrast to characteristics of hardware? APR/MAY 2016** ○ Software is easier to change than hardware. The cost of change is much higher for hardware than for software. ○ Software products evolve through multiple releases by adding new features and re-writing existing logic to support the new features. Hardware products consist of physical components that cannot be “refactored” after manufacturing, and cannot add new capabilities that require hardware changes. ○ Specialized hardware components can have much longer lead times for acquisition than is true for software. ○ Hardware design is driven by architectural decisions. More of the architectural work must be done up front compared to software products. ○ The cost of development for software products is relatively flat over time.	C215.1	**BTL5**

	However, the cost of hardware development rises rapidly towards the end of the development cycle. ○ Testing software commonly requires developing thousands of test cases. Hardware testing involves far fewer tests. Hardware must be designed and tested to work over a range of time and environmental conditions, which is not the case for software.		
15	**List the process maturity levels in SEIs CMM. <u>NOV/DEC2015</u>** Level 1:Initial– Few processes are defined and individual efforts are taken. Level 2:Repeatable– To track cost schedule and functionality basic project management processes are established. Level 3:Defined– The process is standardized, documented and followed. Level 4:Managed– Both the software process and product are quantitatively understood and controlled using detailed measures.	C215.1	**BTL6**
16	**What does Verification represent?** Verification represents the set of activities that are carried out to confirm that the software correctly implements the specific functionality.	C215.1	**BTL6**
17	**What does Validation represent?** Validation represents the set of activities that ensure that the software that has been built is satisfying the customer requirements.	C215.1	**BTL5**
18	**What are the steps followed in testing? <u>MAY/JUNE 2016</u>** i. Unit testing – The individual components are tested in this type of testing. ii. Module testing – Related collection of independent components are tested. iii. Sub-system testing – This is a kind of integration testing. Various modules are integrated into a subsystem and the whole subsystem is tested. iv. System testing – The whole system is tested in this system. v. Acceptance testing – This type of testing involves testing of the system with customer data.If the system behaves as per customer need then it is accepted.	C215.1	**BTL6**

19	**State the advantages and disadvantages in LOC based cost estimation? APR/MAY 2015** **Advantages of LOC** □ It is straight forward (simple) □ Easily can be automated (plenty of tools are available) **Disadvantages of LOC** □ Its Language dependent □ Penalizes the well designed short programs □ Cannot easily accommodate nonprocedural languages □ Need a level of detail that may not be available at the early stages of development.	C215.1	**BTL6**
20	**What is requirement engineering?** Requirement engineering is the process of establishing the services that the customer requires from the system and the constraints under which it operates and is developed.	C215.1	**BTL6**
21	**What are the various types of traceability in software engineering?** i. Source traceability – These are basically the links from requirement to stakeholders who propose these requirements. ii. Requirements traceability – These are links between dependant requirements. iii. Design traceability – These are links from requirements to design.	C215.1	**BTL6**
22	**If you have to develop a word processing software product, what process models will you choose? Justify your answer. NOV/DEC 2016** We will choose the incremental model for word processing software. It focuses on	C215.1	**BTL5**

	the aspects of the word processing software that are visible to the customer / end user. The feedback is used to refine the prototype.		
23	**What led to the transition from product to process oriented development in software engineering? APR/MAY 2016** Product techniques to designing software - Large numbers of software projects do not meet their expectations in terms of functionality, cost, or delivery schedule. Process - Composed of line practitioners who have varied skills, the group is at the center of the collaborative effort of everyone in the organization who is involved with software engineering process improvement. *Process-oriented* view on cooperating software components based on the concepts and terminology of a language/action perspective on cooperative work provides a more suitable foundation for the analysis, design and implementation of software components in business applications.	C215.1	**BTL6**
24	**What are the advantages and disadvantages of iterative software development model NOV/DEC 2015** Advantages • In iterative model we can only create a high-level design of the application before we actually begin to build the product and define the design solution for the entire product. • Building and improving the product step by step. • can get the reliable user feedback • Less time is spent on documenting and more time is given for designing. Disadvantages • Each phase of an iteration is rigid with no overlaps • Costly system architecture or design issues may arise because not all requirements are gathered up front for the entire lifecycle	C215.1	**BTL6**

25	**What are the issues in measuring the software size using LOC as metric** **NOV/DEC 2015, NOV/DEC 2017** • Lack of Accountability. • Lack of Cohesion with Functionality. • Adverse Impact on Estimation. • Difference in Languages. • Advent of GUI Tools • Lack of Counting Standards.	C215.1	**BTL5**
26	**What is System Engineering? April/may 2018** System Engineering means designing, implementing, deploying and operating systems which include hardware, software and people.	C215.1	**BTL6**
27	**What is the use of CMM? NOV/DEC2015** Capability Maturity Model is used in assessing how well an organization's processes allow to complete and manage new software projects.	C215.1	**BTL6**
28	**What is meant by Software engineering paradigm?** The development strategy that encompasses the process, methods and tools and generic phases is often referred to as a process model or software engineering paradigm.	C215.1	**BTL6**
29	**Define agility and agile team.** **April /May 2015** ■ Agility-Effective (rapid and adaptive) response to change (team members, new technology, requirements) ■ Effective communication in structure and attitudes among all team members, technological and business people, software engineers and managers。 ■ Drawing the customer into the team. Eliminate "us and them" attitude. Planning in an uncertain world has its limits and plan must be flexible. ■ **Organizing a team** so that it is in control of the work performed ■ The development guidelines stress delivery over analysis and design although these activates are not discouraged, and active and continuous	C215.1	**BTL5**

	communication between developers and customers ■ Eliminate all but the most essential work products and keep them lean. Emphasize an incremental delivery strategy as opposed to intermediate products that gets working software to the customer as rapidly as feasible		
30	**Write any two characteristics of software as a product. <u>April /May 2015</u>** 1. Software is developed or engineered, it is not manufactured in the classical sense 2. Software doesn't "wear out." 3. Although the industry is moving toward component-based assembly, most software continues to be custom built.	C215.1	**BTL6**
31	**Write the IEEE definition of software engineering . NOV/DEC 2017** According to **IEEE's definition software engineering** can be **defined** as the application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of **software**, and the study of these approaches; that is, the application of **engineering** to **software**.	C215.1	**BTL6**
32	**List two deficiencies in waterfall model . Which process model do you suggest to overcome each deficiency. APRIL/MAY 2017** •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. • No working software is produced until late during the life cycle.	C215.1	**BTL6**
33	**What is Agile?** The word 'agile' means − • Able to move your body quickly and easily. • Able to think quickly and clearly. In business, 'agile' is used for describing ways of planning and doing work wherein it is understood that making changes as needed is an important part of the job. Business'agililty' means that a company is always in a position to take account of the market changes. In software development, the term 'agile' is adapted to mean 'the ability to respond to changes − changes from Requirements, Technology and People.'	C215.1	**BTL6**

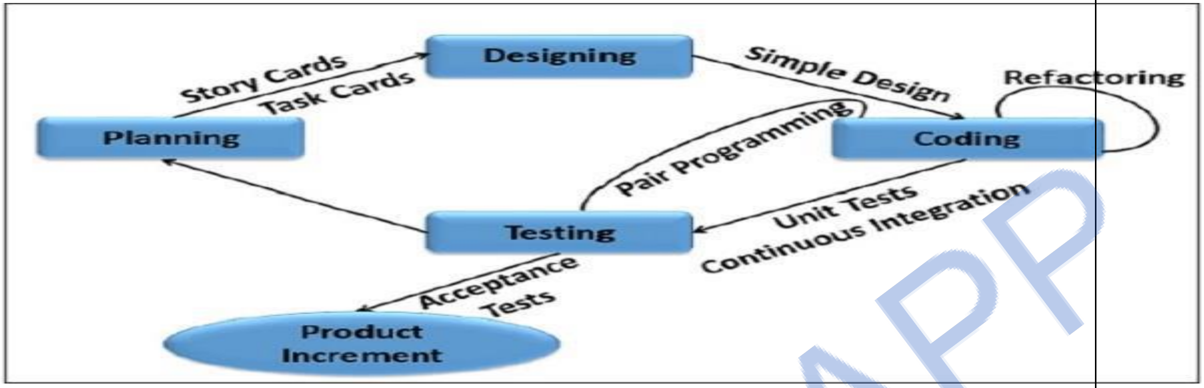
34	**What is Agile Manifesto?** The Agile Manifesto states that – We are uncovering better ways of developing software by doing it and helping others do it. Through this work, we have come to value – • **Individuals and interactions** over processes and tools. • **Working software** over comprehensive documentation. • **Customer collaboration** over contract negotiation. • **Responding to change** over following a plan. That is, while there is value in the items on the right, we value the items on the left more.	C215.1	**BTL5**
35	**What are the Characteristics of Agility?** following are the characteristics of Agility – • Agility in Agile Software Development focuses on the culture of the whole team with multi-discipline, cross-functional teams that are empowered and selforganizing. • It fosters shared responsibility and accountability. • Facilitates effective communication and continuous collaboration. • The whole-team approach avoids delays and wait times. • Frequent and continuous deliveries ensure quick feedback that in in turn enable the team align to the requirements. • Collaboration facilitates combining different perspectives timely in implementation, defect fixes and accommodating changes.	C215.1	**BTL6**
36	**What are the principles of of agile methods?** **Customer involvement** Customers should be closely involved throughout the development process. Their role is provide and prioritize new system requirements and to evaluate the iterations of the	C215.1	**BTL6**

	system. **Incremental delivery** The software is developed in increments with the customer specifying the requirements to be included in each increment. **People not process** The skills of the development team should be recognized and exploited. Team members should be left to develop their own ways of working without prescriptive processes. **Embrace change** Expect the system requirements to change and so design the system to accommodate these changes. **Maintain simplicity** Focus on simplicity in both the software being developed and in the development process. Wherever possible, actively work to eliminate complexity from the system.		
37	**What are the Problems with agile methods?** • It can be difficult to keep the interest of **customers** who are involved in the process. • **Team members** may be unsuited to the intense involvement that characterizes agile methods. • Prioritizing changes can be difficult where there are **multiple stakeholders**. • **Maintaining simplicity** requires extra work. • **Contracts** may be a problem as with other approaches to iterative development.	C215.1	**BTL6**
38	**What is Extreme Programming?** XP is a lightweight, efficient, low-risk, flexible, predictable, scientific, and fun way to develop a software. e**X**treme **P**rogramming (XP) was conceived and developed to address the specific needs of software development by small teams in the face of vague and changing requirements. Extreme Programming is one of the Agile software development methodologies. It provides values and principles to guide the team behavior. The team is expected to self-organize. Extreme Programming provides specific core practices where – • Each practice is simple and self-complete. • Combination of practices produces more complex and emergent behavior.	C215.1	**BTL5**

39	**HOW Embrace Change happens in Extreme programming?** A key assumption of Extreme Programming is that the cost of changing a program can be held mostly constant over time. This can be achieved with − • Emphasis on continuous feedback from the customer • Short iterations • Design and redesign • Coding and testing frequently • Eliminating defects early, thus reducing costs • Keeping the customer involved throughout the development • Delivering working product to the customer	C215.1	**BTL6**
40	**How Extreme Programming usedin a Nutshell?** Extreme Programming involves − • Writing unit tests before programming and keeping all of the tests running at all times. The unit tests are automated and eliminates defects early, thus reducing the costs. • Starting with a simple design just enough to code the features at hand and redesigning when required. • Programming in pairs (called pair programming), with two programmers at one screen, taking turns to use the keyboard. While one of them is at the keyboard, the other constantly reviews and provides inputs. • Integrating and testing the whole system several times a day.	C215.1	**BTL5**
41	**Why is it called "Extreme?** Extreme Programming takes the effective principles and practices to extreme levels.	C215.1	**BTL6**

	• Code reviews are effective as the code is reviewed all the time. • Testing is effective as there is continuous regression and testing. • Design is effective as everybody needs to do refactoring daily. • Integration testing is important as integrate and test several times a day. • Short iterations are effective as the planning game for release planning and iteration planning.		
42	**What are the Extreme Programming Advantages?** Extreme Programming solves the following problems often faced in the software development projects − • **Slipped schedules** − and achievable development cycles ensure timely deliveries. • **Cancelled projects** − Focus on continuous customer involvement ensures transparency with the customer and immediate resolution of any issues. • **Costs incurred in changes** − Extensive and ongoing testing makes sure the changes do not break the existing functionality. A running working system always ensures sufficient time for accommodating changes such that the current operations are not affected. • **Production and post-delivery defects: Emphasis is on** − the unit tests to detect and fix the defects early.	C215.1	**BTL5**
43	**What is Scrum ?** The Scrum approach is a general agile method but its focus is on managing iterative development rather than specific agile practices. There are three phases in Scrum: 1. The initial phase is an outline planning phase where you establish the general objectives for the project and design the software architecture. 2. This is followed by a series of **sprint** cycles, where each cycle develops an	C215.1	**BTL6**

	increment of the system. 3. The project closure phase wraps up the project, completes required documentation such as system help frames and user manuals and assesses the lessons learned from the project.		
44	**What are the** Advantages **of scrum ?** • The product is broken down into a set of **manageable and understandable chunks**. • Unstable requirements do not hold up **progress**. • The whole team have visibility of everything and consequently **team communication** is improved. • Customers see **on-time delivery** of increments and gain feedback on how the product works. • **Trust** between customers and developers is established and a positive culture is created in which everyone expects the project to succeed.	C215.1	**BTL6**
45.	**Mention the Two perspectives on scaling of agile methods?** 1. **Scaling up** 2. **Scaling out**	C215.1	**BTL6**
46.	**What is Scaling up** Using agile methods for developing large software systems that cannot be developed by a small team. For large systems development, it is not possible to focus only on the code of the system; you need to do more up-front design and system documentation. Cross-team communication mechanisms have to be designed and used, which should involve regular phone and video conferences between team members and frequent, short electronic meetings where teams update each other on progress. Continuous integration, where the whole system is built every time any developer checks in a change, is practically impossible; however, it is essential to maintain frequent system builds and regular releases of the system.	C215.1	**BTL5**
47.	**What isScaling out.** How agile methods can be introduced across a large organization with many years of software development experience. Project managers who do not have experience of agile methods may be reluctant to accept the risk of a new approach. Large organizations often have quality procedures and standards that all projects are expected to follow and, because of their bureaucratic nature, these are likely to be incompatible with agile methods. Agile methods seem to work best when team members have a relatively high skill level. However, within large organizations, there are likely to be a wide range of skills and abilities.	C215.1	**BTL6**

48.	**Draw the diagram of Extreme programming?**  Designing Story Cards Task Cards Simple Design Refactoring Planning Pair Programming Coding Testing Unit Tests Continuous Integration Acceptance Tests Product Increment	C215.1	**BTL6**
49	**What is agile development?** Specification, design, implementation and testing are inter-leaved and the outputs from the development process are decided through a process of negotiation during the software development process. projects include elements of plan-driven and agile processes. Deciding on the balance depends on many technical, human, and organizational issues.	C215.1	**BTL5**
50.	**What is Scrum master?** The role of the ScrumMaster is to protect the development team from external distractions. At the end of the sprint the work done is reviewed and presented to stakeholders (including the product owner).	C215.1	**BTL6**

PART –B

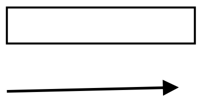
S.NO	QUESTIONS	CO	BLOOM’S LEVEL
1	**Explain the following: (i) waterfall model (ii) Spiral model (iii)RAD model (iv) Prototyping model. NOV/DEC-12,**	C215.1	**BTL6**

	NOV/DEC-15, Press-Pg-no – 79,86,81,83		
2	**Discuss in detail the project structure and programming team structure of a software organization. NOV/DEC-10** Press-Pg-no – 68	C215.1	**BTL5**
3	**Discuss the various life cycle models in software development? APR/MAY-16** Press-Pg-no-77	C215.1	**BTL6**
4	**What is the difference between information engineering & product engineering? Also explain the product engineering hierarchy in detail. MAY/JUN-13** Press-Pg-no- 161	C215.1	**BTL6**
5	**Write note on business process engineering and product engineering? MAY/JUN-13 , APRIL/MAY-15** Press-Pg-no- 161	C215.1	**BTL5**
6	**Explain in detail about spiral model with a neat sketch and describe why this model comes under both evolutionary and RAD models. APRIL/MAY-15, NOV/DEC 2017** Press-Pg-no- 186	C215.1	**BTL6**
7	**Which process model is best suited for risk management? Discuss in detail with an example. Give its advantages and disadvantages? NOV/DEC 2016,APRIL/MAY 2018** Press-Pg-no – 93	C215.1	**BTL6**
8	**(a) List the principles of agile software development. NOV/DEC 2016** Press-Pg-no – 67 **(b) Consider 7 functions with their estimated lines of code. Average productivity based on historical data is 620 LOC/pm and labour rate is Rs. 8000 per mnth. Find the total estimates project cost and effort? F1 – 2340 , F2 – 5380, F3 – 6800 , F4 –**	C215.1	**BTL5**

	3350 , F5 -4950 , F6 -2140 , F7 – 8400 Refer class notes.		
9	**(i) What is the impact of reusability in software development process?** **(ii) Explain the component based software development model with a neat sketch. NOVDEC 2017** Refer class notes	C215.1	**BTL6**
10	**(i)How function point analysis methodology is applied in estimation of software size ?Explain. Why FPA methodology is better than LOC methodology ?** **(ii)An application has the following:10 low external inputs, 12 high external outputs, 20 low internal logical files, 15 high external interface files, 12 average external inquiries and a value adjustment factor of 1.10 . What is the unadjusted and adjusted function point count ? APRIL/MAY 2017** Refer class notes	C215.1	**BTL5**
11	**What is a process model ? Describe the process model that you would choose to manufacture a car. Explain giving suitable reasons. APRIL/MAY 2017** Refer class notes	C215.1	**BTL6**
12	**Explain how breakdown structure is used in software engineering .Discuss how software project scheduling helps in timely release of a product. APRIL/MAY 2018** Refer class notes	C215.1	**BTL5**
13	**Give detail explanation about agile process?** Refer class notes	C215.1	**BTL6**
14	**Describe in detail about Extreme programming ?** Refer class notes	C215.1	**BTL5**
15	**Explain about Extreme Programming using nutshell.?** Refer class notes	C215.1	**BTL6**

UNIT – 2

PART –A

S.NO	QUESTIONS	CO	BLOOM’S LEVEL
1	**What is Software Prototyping? <u>NOV/DEC-10 , APR/MAY-11, MAY/JUNE-13</u>** It is a rapid software development for validating the requirements. It is to help customers & developers to understand the system requirements.	C215.2	**BTL6**
2	**Define functional and non- Functional requirements. <u>NOV/DEC-10</u>** Functional requirements describe all the functionality or system services. It should be clear how system should react to particular inputs and how particular systems behave in particular situation. Non functional requirements define the system properties and constraints. It is divided in to product, organizational & external requirements.	C215.2	**BTL6**
3	**What is meant by functional requirement? <u>APR/MAY-11</u>** Functional requirements describe all the functionality or system services. It should be clear how system should react to particular inputs and how particular systems behave in particular situation.	C215.2	**BTL6**
4	**Name the metrics for specifying Non-functional requirements? <u>NOV/DEC-11</u>** Speed, size, ease of use, reliability, robustness, portability	C215.2	**BTL3**
5	**Draw the DFD for the following (i) External entity (ii) Data items <u>NOV/DEC-11</u>** External entity Data items 	C215.2	**BTL2**

6	**What do requirements processes involve? APR/MAY-12** It involves feasibility study, discovery, analysis &validation of system requirements.	C215.2	**BTL5**
7	**Define non-functional requirements. APR/MAY-12** Non functional requirements define the system properties and constraints. It is divided in to product, organizational & external requirements	C215.2	**BTL6**
8	**Distinguish between the term inception, elicitation, & elaboration with reference to requirements? NOV/DEC-12** Inception – set of questions are asked to establish basic understanding of problem. Elicitation - collaborative requirements gathering & quality function deployment Elaboration – It focuses on developing a refined technical model of software function, features & constraints.	C215.2	**BTL6**
9	**An SRS is traceable ?comment NOV/DEC-12,MAY/JUNE 2016** An SRS is correct if, and only if, every requirement stated therein is one that the software shall meet. Traceability makes this procedure easier and less prone to error.	C215.2	**BTL2**
10	**What is data dictionary? MAY/JUN-13 , APR/MAY 2016 , NOV/DEC 2016, APRIL/MAY 2017** It is organized collection of all the data elements of the system with precise and rigorous definition so that user & system analyst will have a common understanding of inputs, outputs, components of stores and intermediate calculations.	C215.2	**BTL6**
11	**What are the benefits of prototyping?** i. Prototype serves as a basis for deriving system specification. ii. Design quality can be improved. iii. System can be maintained easily.	C215.2	**BTL6**

	iv. Development efforts may get reduced. v. System usability can be improved.		
12	**What are the prototyping approaches in software process?<u>MAY/JUNE 2016,APRIL/MAY 2018</u>** i. Evolutionary prototyping – In this approach of system development, the initial prototype is prepared and it is then refined through number of stages to final stage. ii. Throw-away prototyping – Using this approach a rough practical implementation of the system is produced. The requirement problems can be identified from this implementation. It is then discarded.System is then developed using some different engineering paradigm.	C215.2	**BTL6**
13	**List the characteristics of good SRS? <u>APR/MAY 2016</u>** • Correct • Unambiguous • Complete • Consistent • Ranked for importance and/or stability • Verifiable • Modifiable • Traceable	C215.2	**BTL6**
14	**Classify the following as functional / non-functional requirements for a banking system? <u>NOV / DEC 2016</u>** (a) Verifying bank balance – **functional requirements** (b) Withdrawing money from bank – **functionalrequirements** (c) Completion of transaction in less than 1 sec – non-**functional requirements** (d) Extending system by providing more tellers for customers -	C215.2	**BTL6**

	non-functional requirements		
15	**What is the linkage between Dataflow and ER diagram?APR/MAY 2016** An ER diagram is the Entity Relationship Diagram, showing the relationship between different entities in a process. A Data Flow diagram is a symbolic structure showing how the flow of data is used in different process	C215.2	**BTL6**
16	**List the steps in user interface design? Golden rules of UI design APR/MAY 2015, NOV/DEC2015** Place the User in Control Reduce the User's Memory Load Make the Interface Consistent	C215.2	**BTL6**
17	**How are requirements validated?APR/MAY 2015** **Requirements validation:** Have we got the requirements right? In the validation phase, the work products produced as a consequence of requirements engineering are examined for consistency, omissions, and ambiguity. The basic objective is to ensure that the SRS reflects the actual requirements accurately and clearly.	C215.2	**BTL6**
18	**What is a state transition diagram?** State transition diagram is basically a collection of states and events. The events cause the system to change its state. It also represents what actions are to be taken based on the transition.	C215.2	**BTL2**
19	**What is DFD?** Data Flow Diagram depicts the information flow and the transforms that are applied on the data as it moves from input to output.	C215.2	**BTL3**
20	**What is waterfall model?** The Waterfall Model was first Process Model to be introduced. It is also referred to as a **linear-sequential life cycle model**.	C215.2	**BTL3**

	It is very simple to understand and use. In a waterfall model, each phase must be completed fully before the next phase can begin. This type of model is basically used for the for the project which is small and there are no uncertain requirements. In this model the testing starts only after the development is complete. In **waterfall model phases** do not overlap.		
21	**What is ERD?** Entity Relationship Diagram is the graphical representation of the object relationship pair. It is mainly used in database applications.	C215.2	**BTL6**
22	**What is data modeling?** Data modeling is the basic step in the analysis modeling. In data modeling the data objects are examined independently of processing. The data model represents how data are related with one another.	C215.2	**BTL2**
23	**What is requirement engineering?** Requirement engineering is the process of establishing the services that the customer requires from the system and the constraints under which it operates and is developed.	C215.2	**BTL6**
24	**What are the various Rapid prototyping techniques? <u>April /May 2015</u>** i. Dynamic high level language development. ii. Database programming. iii. Component and application assembly.	C215.2	**BTL6**
25	**What is data modeling?** Data modeling is the basic step in the analysis modeling. In data modeling the data objects are examined independently of processing. The data model represents how data are related with one another.	C215.2	**BTL6**

26	**What are the various types of traceability in software engineering? April/may 2018** i. Source traceability – These are basically the links from requirement to stakeholders ii. Requirements traceability – These are links between dependant requirements. iii. Design traceability – These are links from requirements to design.	C215.2	**BTL6**
27	**What is cardinality in data modeling?** Cardinality in data modeling, cardinality specifies how the number of occurrences of one object is related to the number of occurrences of another object.	C215.2	**BTL5**
28	**What are the objectives of Analysis modeling?** i. To describe what the customer requires. ii. To establish a basis for the creation of software design. iii. To devise a set of valid requirements after which the software can be built.	C215.2	**BTL6**
29	**How the limitations of waterfall model overcome? April /May 2015** This type of model is basically used for the for the project which is small and there are no uncertain requirements.Where no overlapping of phases. At the end of each phase, a review takes place to determine if the project is on the right path and whether or not to continue or discard the project.	C215.2	**BTL6**
30	**What is feasibility study? NOV/DEC2015 , APR/MAY 2016** software feasibility has four solid dimensions: Technology— Is a project technically feasible? Is it within the state of the art? Can defects be reduced to a level matching the application's needs? Finance—Is it financially feasible? Can development be completed	C215.2	**BTL6**

	at a cost the software organization, its client, or the market can afford? Time—Will the project's time-to-market beat the competition? Resources—Does the organization have the resources needed to succeed? Before starting any project the feasibility study team ought to carry initial architecture and design of the high-risk requirements to the point at which it can answer these questions. In some cases, when the team gets negative answers, a reduction in requirements may be negotiated.		
31	**Define Quality function decelopment(QFD). NOV/DEC 2017** **Quality Function** Deployment (**QFD**) is a structured approach to defining customer needs or requirements and translating them into specific plans to produce products to meet those needs. The "voice of the customer" is the term to describe these stated and unstated customer needs or requirements.	C215.2	**BTL6**
32	**Differentiate between normal and exciting requirements ? APRIL/MAY 2017** **Normal requirements** • The objective and goal are stated for the system through the meetings with the customer. • For the customer satisfaction these requirements should be there. **Exciting requirements** • These features are beyond the expectation of the customer. • The developer adds some additional features or unexpected feature into the software to make the customer more satisfied. **For example,** the mobile phone with standard features, but	C215.2	**BTL6**

	the developer adds few additional functionalities like voice searching, multi-touch screen etc. then the customer more exited about that feature.		
33	**How do you design a software project for reuse? (Nov/Dec 2007)** • A clear and well-defined product vision is an essential foundation to an software project. • An evolutionary implementation strategy would be a more pragmatic strategy for the company. • There exist a need for continuous management support and leadership to ensure success.	C215.2	**BTL6**
34	**What are the standards for documentation? Briefly explain (Nov/Dec 2007)** IEEE Std 1028-2008 This standard defines five types of software reviews and procedures for their execution. Review types include management reviews, technical reviews, inspections, walk-throughs and audits. IEEE Std 1012-2004 This standard describes software verification and validation processes that are used to determine if software products of an activity meets the requirements of the activity and to determine if software satisfies the user's needs for the intended usage. The scope includes analysis, evaluation, review, inspection, assessment and testing of both products and processes.	C215.2	**BTL6**
35	**What are context free questions? How it differs from meta questions? (Nov/Dec 2009)** Context free questions are questions that can be used regardless of the project under consideration. They are general questions about the nature of the project and the environment in which the final product will be used.Meta questions are very complex and detailed questions about the project model	C215.2	**BTL6**

36	**Define behaviouralmodelling(Nov/Dec 2012)** **All behavioural models really do is describe the control structure of a system.** **This can be things like:** ▪ **Sequence of operations** ▪ **Object states** ▪ **and Object interactions** **Furthermore, this modelling layer can also be called Dynamic Modelling. The activity of creating a behavioural model is commonly known as behavioural modelling. As well as this, a system should also only have one behavioural model – much like functional modelling.**	C215.2	**BTL6**
37	**what are the types of prototypes** • Evolutionary prototyping – the initial prototype is prepared and it is then refined through number of stages to final stage. • Throw-away prototyping – a rough practical implementation of the system is produced. The requirement problems can be identified from this implementation	C215.2	**BTL6**
38	**Define behaviouralmodelling(Nov/Dec 2012)** **All behavioural models really do is describe the control structure of a system.** **This can be things like:** ▪ **Sequence of operations** ▪ **Object states** ▪ **and Object interactions** **Furthermore, this modelling layer can also be called Dynamic Modelling. The activity of creating a behavioural model is commonly known as behavioural modelling. As well as this, a system should also only have one behavioural model – much like functional modelling.**	C215.2	**BTL6**
39	**What is the major distinction between user requirement and system requirement? (April/May 2008)** User requirements may be a set of statements or use case scenarios presented by the client in layman's terms of which the client can easily	C215.2	**BTL6**

	elaborate and are usually free of technical jargon. System requirements are built from the clients input being what they have specified in the user requirements.		
40	**Which style of prototyping is most appropriate when the requirement are not well-understood? (April/May 2008)** User Interface prototyping is most appropriate.This prototyping is used to prespecify the look and feel of user interface in an effective way.	C215.2	**BTL6**
41	**Specify at least four questionnaire which supports to select the prototyping approach. (Nov/Dec 2009)** • Prototype serves as a basis for deriving system specification. • Design quality can be improved. • System can be maintained easily. • Development efforts may get reduced. • System usability can be improved.	C215.2	**BTL6**
42	**What is the purpose of domain analysis. (April/May 2010)** Domain analysis, or product line analysis, is the process of analysing related software systems in a domain to find their common and variable parts. It is a model of wider business context for the system	C215.2	**BTL3**
43	**what are the types of prototypes** • Evolutionary prototyping – the initial prototype is prepared and it is then refined through number of stages to final stage. • Throw-away prototyping – a rough practical implementation of the system is produced. The requirement problems can be identified from this implementation	C215.2	**BTL6**
44	**list two advantage of employing prototyping in software process?** • Prototype serves as a basis for deriving system specification. • Design quality can be improved. • System can be maintained easily. • Development efforts may get reduced. • System usability can be improved.	C215.2	**BTL6**

45	**State the different criteria applied to evaluate an effective modular system.** **(May/June 2006)** • A system is considered modular if it consists of discreet components so that each component can be implemented separately, and a change to one component has minimal impact on other components. • Modularity is a clearly a desirable property in a system. Modularity helps in system debugging. Isolating the system problem to a component is easier if the system is modular.	C215.2	**BTL6**
46	**What is meant by structural analysis?** The structural analysis is mapping of problem domain to flows and transformations. The system can be modeled by using Entity Relationship diagram, Data flow diagram and Control flow diagrams.	C215.2	**BTL2**
47	**What is the outcome of feasibility study?** The outcome of feasibility study is the results obtained from the following questions: x Which system contributes to organizational objectives? x Whether the system can be engineered? Is it within the budget? x Whether the system can be integrated with other existing system?	C215.2	**BTL3**
48	**What are nonfunctional requirements?** Nonfunctional requirements are constraints on the services or functions offered by the system such as timing constraints, constraints on the development process, standards, etc…	C215.2	**BTL6**
49	**What are the advantages of evolutionary prototyping?** i. Fast delivery of the working system. ii. User is involved while developing the system. iii. More useful system can be delivered. iv. Specification, design and implementation work in co-ordinate manner.	C215.2	**BTL5**
50	**What are the various Rapid prototyping techniques?** i. Dynamic high level language development. ii. Database	C215.2	**BTL6**

	programming. iii. Component and application assembly.		

PART –B

S.NO	QUESTIONS	CO	BLOOM'S LEVEL
1	**Discuss any four process models with suitable application. NOV/DEC-10 , APR/MAY-11, NOV/DEC-12, MAY/JUN-13** *Somm-Pg-no- 164*	C215.2	**BTL6**
2	**Explain the execution of seven distinct functions accomplished inrequirement engineering process / Explain briefly the requirement engineering process with neat sketch and describe each process with an example. APRIL/MAY-15 NOV/DEC-15, NOV/DEC 2017, APRIL/MAY 2017** *Press-Pg-no- 176*	C215.2	**BTL6**
3	**What is data dictionary? Explain. How to select the appropriate prototyping approach?APR/MAY-11, APR/MAY-12, NOV/DEC2015** *Refer class notes.*	C215.2	**BTL6**
4	**How does the analysis modeling help to capture unambiguous & consistent requirements? Discuss several methods for requirements validation? NOV/DEC-11** *Press-Pg-no- 211*	C215.2	**BTL5**
5	**Explain prototyping in the software process.APRIL/MAY-15 MAY/JUNE 2016** Press-pg no –229	C215.2	**BTL6**
6	**Explain the functional & behavioral model for software**	C215.2	**BTL5**

	requirements process? NOV/DEC-12, MAY/JUN-13,NOV/DEC 2013 *Press-Pg-no- 226*		
7	**Explain metrics for specifying non-functional requirements? IEEE standarad software requirement document? MAY/JUN-13** *Somm-Pg-no- 141,158*	C215.2	**BTL6**
8	**What is requirements elicitation? Explain various activities performed in it with watch system that facilitates to set time and alarm as an example? NOV/DEC 2016, APRIL/MAY 2017, APRIL/MAY 2018** Press-Pg-no – 168	C215.2	**BTL6**
9	**What is the purpose of data flow diagrams? What are the notations used for the same. Explain by constructing a context flow diagram level -0 DFD and level-1 DFD for a library management system? NOV/DEC 2016** Press-Pg-no – 284	C215.2	**BTL5**
10	**Consider the process of ordering a pizza over the phone. Draw the use case diagram and also sketch the activity diagram representing each step of the process, from the moment you pick up the phone to the point where you start eating the pizza. Include activities that others need to perform. Add exception handling to the activity diagram you developed. Consider at least two exceptions.(Ex : Delivery person wrote down wrong address, deliver person brings wrong pizza). NOV/DEC 2017** Refer class notes.	C215.2	**BTL6**
11	**Explain the feasibility studies. What are the outcomes? Does it have implicit or explicit effects on software requirement collection. APRIL/MAY 2017**	C215.2	**BTL5**
12	**What is SRS?Explain in detail about various component of an**	C215.2	**BTL6**

	SRS.		
13	**What is requirement engineering? State its process and explain requirement elicitation problem. (April/May 2008)** Refer class notes	C215.2	**BTL6**
14	**what is prototyping .explain its types types.(Nov/Dec 2009)** Refer notes	C215.2	**BTL6**
15	An Independent Truck Company Wants To Track And Record Its Drivers Driving Habits. For This Purpose The Company Has Rented 800 Phone Numbers And Has Printed The Numbers On The Front, Back And Side Of All Trucks Owned By The Company.Next To The 800 Numbers A Message Is Written”Plese Report Any Driver Of Truck Problem By Calling This Number**”(Nov/Dec 2012**) Refer class notes.	C215.2	**BTL5**

<u>UNIT – 3</u>

PART –A

S.NO	QUESTIONS	CO	BLOOM'S LEVEL
1	**What are the primary interaction styles and state their advantages? NOV/DEC-10** 1.Direct manipulation - Easiest to grasp with immediate feedback , Difficult to program 2. Menu selection - User effort and errors minimized, large numbers and combinations of choices a problem 3. Form fill-in - Ease of use, simple data entry, Tedious, takes a lot of screen space 4. Command language - Easy to program and process, Difficult to master for casual users 5. Natural language - Great for casual users, Tedious for expert users.	C215.3	**BTL5**
2	**List the architectural models that can be developed. NOV/DEC-10** Data-centered architectures, Data flow architectures, Call and return architectures Object-oriented architectures, Layered architectures.	C215.3	**BTL5**
3	**What is meant by real time system design? APR/MAY-11** A real-time system is a software system where the correct functioning of the system depends on the results produced by the system and the time at which these results are produced.	C215.3	**BTL3**
4	**List four design principles of a good design? APR/MAY-11APRIL/MAY 2018**	C215.3	**BTL5**

	○ Process should not suffer from tunnel vision. ○ It should be traceable to the analysis model ○ It should not reinvent the wheel ○ It should exhibit uniformity & integration.		
5	**List out design methods. APR/MAY-12** Architectural design , data design , modular design.	C215.3	**BTL5**
6	**Define data acquisition APR/MAY-12,MAY/JUN-13** Collect data from sensors for subsequent processing and analysis.	C215.3	**BTL4**
7	**How do you apply modularization criteria for a monolithic software NOV/DEC-12** Modularity is achieved to various extents by different modularization approaches. Code based modularity allows developers to reuse and repair parts of the application, but development tools are required to perform these maintenance functions .Object based modularity provides the application as a collection of separate executable files which may be independently maintained and replaced without redeploying the entire application.	C215.3	**BTL5**
8	**What is the design quality attributes 'FURPS' meant? NOV/DEC-12, NOV/DEC2015, NOV/DEC2017** FURPS is an acronym representing a model for classifying software quality attributes (functional and non-functional requirements) Functionality, Usability, Reliability, Performance and Supportability model.	C215.3	**BTL5**
9	**Define data abstraction? MAY/JUN-13** Data abstraction is a named collection of data that describes the data object. Eg:- Door attribute – door type, swing direction, weight	C215.3	**BTL5**

10	**What are the elements of design model?** i. Data design ii. Architectural design iii. Interface design iv. Component-level design	C215.3	**BTL5**
11	**What is the benefit of modular design?** Changes made during testing and maintenance becomes manageable and they do not affect other modules.	C215.3	**BTL5**
12	**Name the commonly used architectural styles.** i. Data centered architecture. ii. Data flow architecture. iii. Call and return architecture. iv. Object-oriented architecture. v. Layered architecture.	C215.3	**BTL5**
13	**What is a cohesive module?** A cohesive module performs only “one task” in software procedure with little interaction with other modules. In other words cohesive module performs only one thing.	C215.3	**BTL6**
14	**What are the different types of Cohesion?** i. Coincidentally cohesive –The modules in which the set I\of tasks are related with each other loosely then such modules are called coincidentally cohesive. ii. Logically cohesive – A module that performs the tasks that are logically related with each other is called logically cohesive. iii. Temporal cohesion – The module in which the tasks need to be executed in some specific time span is called temporal cohesive. iv. Procedural cohesion – When processing elements of a module are related with procedural cohesive. v. Communicational cohesion – When the processing elements of a module share the data then such module is called communicational cohesive.	C215.3	**BTL5**

15	**What is Coupling?What are the various types of coupling** **APRIL/MAY-15,** Coupling is the measure of interconnection among modules in a program structure. It depends on the interface complexity between modules. i. Data coupling – The data coupling is possible by parameter passing or data interaction. ii. Control coupling – The modules share related control data in control coupling. iii. Common coupling – The common data or a global data is shared among modules. iv. Content coupling – Content coupling occurs when one module makes use of data or control information maintained in another module.	C215.3	**BTL6**
16	**What are the common activities in design process?** i. System structuring – The system is subdivided into principle subsystems components and communications between these subsystems are identified. ii. Control modeling – A model of control relationships between different parts of the system is established. iii. Modular decomposition – The identified subsystems are decomposed into modules	C215.3	**BTL5**
17	**What are the benefits of horizontal partitioning?** i. Software that is easy to test. ii. Software that is easier to maintain. iii. Propagation of fewer sideeffects. iv. Software that is easier to extend.	C215.3	**BTL5**
18	**What is vertical partitioning? What are the advantages?** Vertical partitioning often called factoring suggests that the control and work should be distributed top-down in program structure. i. These are easy to maintain changes.	C215.3	**BTL6**

	ii. They reduce the change impact and error propagation		
19	**If a module has logical cohesion, what kind of coupling is this module likely to have? APR/MAY 2016** If a module has logical cohesion, then content coupling can be done. In content coupling one module can make use of data or control information maintained in another	C215.3	**BTL5**
20	**Write the best practices for "coding”? APR/MAY 2015, NOV/DEC2015** Best coding practices are a set of informal rules that the software development community has learned over time which can help improve the quality of software. "The first 90% of the code accounts for the first 90% of the development time. The remaining 10% of the code accounts for the other 90% of the development time." The size of a project or program has a significant effect on error rates, programmer productivity, and the amount of management needed.	C215.3	**BTL5**
21	**What architectural styles are preferred for the following system? Why? NOV/DEC2016** (a) Networking – Data centered Architecture (b) Web based systems – Call and return architecture (c) Banking system - Data centered Architecture.	C215.3	**BTL6**
22	**What is DFD?** Data Flow Diagram depicts the information flow and the transforms that are applied on the data as it moves from input to output.	C215.3	**BTL5**
23	**Name the commonly used architectural styles**. i. Data centered architecture. ii. Data flow architecture. iii. Call and return architecture. iv. Object-oriented architecture. v. Layered architecture.	C215.3	**BTL5**

24	**What is ERD?** Entity Relationship Diagram is the graphical representation of the object relationship pair. It is mainly used in database applications.	C215.3	**BTL6**
25	**What UI design patters are used for the following? NOV/DEC 2016, APRIL/MAY 2017, APRIL/MAY 2018** (a) Page layout – interface design (b) Tables - Design (c) Navigation through menus and web pages – design (d) Shopping cart – interface design, task analysis	C215.3	**BTL5**
26	**What are the various elements of data design?** i. Data object – The data objects are identified and relationship among various data objects can be represented using ERD or data dictionaries. ii. Databases – Using software design model, the data models are translated into data structures and data bases at the application level. iii. Data warehouses – At the business level useful information is identified from various databases and the data warehouses are created.	C215.3	**BTL5**
27	**List the guidelines for data design.** i. Apply systematic analysis on data. ii. Identify data structures and related operations. iii. Establish data dictionary. iv. Use information hiding in the design of data structure. v. Apply a library of useful data structures and operations.	C215.3	**BTL6**
28	**What is a Real time system?** Real time system is a software system in which the correct functionalities of the system are dependent upon results produced by the system and the time at which these results are produced	C215.3	**BTL5**
29	**How do you describe software interface? April /May 2015**	C215.3	**BTL5**

	Software interface - the languages and codes that the applications use to communicate with each other and also with the hardware. Three types of interface may have to be defined • Procedural interfaces; • Data structures that are exchanged; • Data representations. The interface describes the behavior of a software component that is obtained by considering only the interactions of that interface and by hiding all other interactions.		
30	**Explain the qualitative criteria for measuring independence? NOV/DEC-11** 1.C**ohesion:** Cohesion is a qualitative indication of the degree to which a module focuses on just one thing. 2. **Coupling:** Coupling is the measure of interconnection among modules in a program structure. It depends on the interface complexity between modules	C215.3	**BTL3**
31	**What is the purpose of a petrinet ? APRIL/MAY 2017** A Petri net, also known as a place/transition (PT) net, is one of several mathematicalmodeling languages for the description of distributed systems. It is a class of discrete event dynamic system. Petri nets offer a graphical notation for stepwise processes that include choice, iteration, and concurrent execution	C215.3	**BTL5**
32	**What is vertical partitioning?** Vertical partitioning often called factoring suggests that the control and work should be distributed top-down in program structure.	C215.3	**BTL2**
33	**What are the benefits of horizontal partitioning?** i. Software that is easy to test. ii. Software that is easier to maintain. iii. Propagation of fewer side effects. iv. Software that is easier to extend.	C215.3	**BTL5**
34	**What are data acquisition systems?**	C215.3	**BTL5**

	Systems that collect data from sensors for subsequent processing and analysis are termed as data acquisition systems. Data collection processes and processing processes may have different periods and deadlines.		
35	**What is interface design?** The interface design describes how the software communicates within itself, with systems that interoperate with it, and with humans who use it.	C215.3	**BTL5**
36	**What are the elements of design model?** Data design ii. Architectural design iii. Interface design iv. Component-level design	C215.3	**BTL5**
37	**What is coupling?** Coupling is the measure of interconnection among modules in a program structure. It depends on the interface complexity between modules.	C215.3	**BTL5**
38	**Define design process**. Design process is a sequence of steps carried through which the requirements are translated into a system or software model.	C215.3	**BTL6**
39	**What is Transform mapping?** The transform mapping is a set of design steps applied on the DFD in order to map the transformed flow characteristics into specific architectural style.	C215.3	**BTL5**
40	**What is component level design?** The component level design transforms structural elements of the software architecture into a procedural description of software components.	C215.3	**BTL5**
41	**What are the objectives of Analysis modeling?** i. To describe what the customer requires. ii. To establish a basis	C215.3	**BTL5**

	for the creation of software design. iii. To devise a set of valid requirements after which the software can be built.		
42	**What are the various types of coupling?** i**Data coupling** – The data coupling is possible by parameter passing or data interaction. ii. **Control coupling** – The modules share related control data in control coupling. iii. **Common coupling** – The common data or a global data is shared among modules. iv**. Content coupling** – Content coupling occurs when one module makes use of data or control information maintained in another module.	C215.3	**BTL6**
43	What does modality in data modeling indicates? Modality indicates whether or not a particular data object must participate in the relationship.	C215.3	**BTL5**
44	**What does Level0 DFD represent?** Level 0 DFD is called as „fundamental system model‟ or „context model‟. In the context model the entire software system is represented by a single bubble with input and output indicated by incoming and outgoing arrows.	C215.3	**BTL5**
45	**What are the elements of design model?** i. Data design ii. Architectural design iii. Interface design iv. Component-level design	C215.3	**BTL5**
46	**What is data modeling?** Data modeling is the basic step in the analysis modeling. In data modeling the data objects are examined independently of processing. The data model represents how data are related with one another.	C215.3	**BTL5**
47	**What is a data object?** Data object is a collection of attributes that act as an aspect,	C215.3	**BTL6**

	characteristic,quality, or descriptor of the object		
48	**What are attributes?** Attributes are the one, which defines the properties of data object.	C215.3	**BTL5**
49	**What is cardinality in data modeling?** Cardinality in data modeling, cardinality specifies how the number of occurrences of one object is related to the number of occurrences of another object.	C215.3	**BTL5**
50	**What is ERD?** Entity Relationship Diagram is the graphical representation of the object relationship pair. It is mainly used in database applications	C215.3	**BTL5**

PART –B

S.NO	QUESTIONS	CO	BLOOM'S LEVEL
1	**Explain the core activities involved in User Interface design process with necessary block diagrams<u>MAY/JUNE 2016 ,NOV/DEC2015, NOV/DEC 2017</u>** *Somm – Pg-no- 398.*	C215.3	**BTL6**
2	**Explain the various modular decomposition and control styles commonly used in any organizational model.<u>MAY/JUNE 2016</u>** *Somm – Pg-no- 274*	C215.3	**BTL5**
3	**Discuss the process of translating the analysis model in to a software design, List the golden rules of user interface design<u>NOV/DEC2015</u>** *Press-Pg-no- 259 , 357*	C215.3	**BTL6**

4	**Explain the basic concepts of software design APR/MAY-11 , NOV/DEC 2017** *Press-Pg-no- 265*	C215.3	**BTL5**
5	**Explain clearly the concept of coupling & cohesion? For each type of coupling give an example of two components coupled in that way?APRIL/MAY 2015, APRIL/MAY 2017, APRIL/MAY 2018** *Press-Pg-no- 335*	C215.3	**BTL5**
6	**Write short notes on Architectural & component design. MAY/JUN-15,NOV/DEC2015** *Somm – Pg-no- 371*	C215.3	**BTL5**
7	**Bring out the necessity of Real-time system design process with appropriate example?APR/MAY-12, MAY/JUNE-13, APRIL/MAY-**15 *Somm – Pg-no- 361 Somm – Pg-no- 357*	C215.3	**BTL6**
8	**What is structured design? Illustrate the structured design process from DFD to structured chart with a case study.NOV/DEC 2016,** Refer class notes	C215.3	**BTL5**
9	**(a) Describe golden rules for interface design NOV/DEC 2016** *Press-Pg-no- 259 , 357* **(b) Explain component level design with suitable example** Refer class notes	C215.3	**BTL5**
10	**What is software architecture ? Describe in detail different types of software architectural styles with illustrations. APRIL/MAY 2017, APRIL/MAY 2018** Refer class notes	C215.3	**BTL5**
11	**What is the purpose of DFD ?What are the compoeenets of DFD? Construct DFD for the following system..** **An online shopping system for xyz provides many services and benefits to its members and staffs. APRIL/MAY 2018**	C215.3	**BTL5**

	Refer class notes		
12	**Describe in detail about architectural styles?** Refer class notes	C215.3	**BTL5**
13	**Describe the concept of cohesion and coupling. State the difference b/w cohesion and coupling with a suitable example. (April/May Apr/May 2008)**	C215.3	**BTL6**
14	**explain transform mapping with suitable example and design steps involved in it.(Nov/Dec 2012)** Refer class notes	C215.3	**BTL5**
15	**Explain the design principlesin detail** Refer class notes	C215.3	**BTL5**

.

UNIT – 4

PART –A

S.NO	**QUESTIONS**	**CO**	**BLOOM'S LEVEL**
1	**What are the characteristics of good tester? NOV/DEC-10,MAY/JUN-13** All tests should be traceable to customer requirements. Tests should be planned long before testing begins. The Pareto principle applies to software testing.	C215.4	**BTL5**
2	**Define software testing?** Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design, and coding.	C215.4	**BTL6**
3	**What are the objectives of testing?** i. Testing is a process of executing a program with the intend of finding an error. ii. A good test case is one that has high probability of finding	C215.4	**BTL6**

	an undiscovered error. iii. A successful test is one that uncovers as an-yet undiscovered error.		
4	**What is integration testing?and What are the approaches of integration testing?<u>APR/MAY-11</u>** In this testing the individual software modules are combined and tested as a group. It occurs after unit testing & before system testing. 1. The non-incremental testing. 2. Incremental testing.	C215.4	**BTL5**
5	**What is regression testing? <u>APR/MAY-15 , NOV/DEC-11,NOV/DEC 2013,</u>** It tends to verify the software application after a change has been made. It seeks to uncover software errors by partially retesting a modified program.	C215.4	**BTL5**
6	**Distinguish between stress and load testing** Stress testing is subjecting a system to an unreasonable load while denying it the resources (e.g., RAM, disc, mips, interrupts, etc.) needed to process that load. Load testing is subjecting a system to a statistically representative (usually) load. The two main reasons for using such loads is in support of software reliability testing and in performance testing. The term "load testing" by itself is too vague and imprecise to warrant use.	C215.4	**BTL5**
7	**Define black box testing? <u>APR/MAY-12,MAY/JUN-13</u>** A black-box tests are used to demonstrate that software functions are operational, that input is properly accepted and output is correctly produced, and that the integrity of external information.	C215.4	**BTL3**
8	**What is boundary condition testing? <u>APR/MAY-12</u>** It is tested using boundary value analysis. (check BVA – 16 mark question)	C215.4	**BTL5**

9	**How is software testing results related to the reliability of software? NOV/DEC-12** Applying fault avoidance, fault tolerance and fault detection for the project helps to achieve reliability of software.	C215.4	**BTL5**
10	**What is big-bang approach? NOV/DEC-12** Big bang approach talks about testing as the last phase of development. All the defects are found in the last phase and cost of rework can be huge.	C215.4	**BTL5**
11	**Why does software fail after it has passed from acceptance testing?APR/MAY 2016** Each acceptance test represents some expected result from the system. Customers are responsible for verifying the correctness of the acceptance tests and reviewing test scores to decide which failed tests are of highest priority. Acceptance tests are also used as regression tests prior to a production release. A user story is not considered complete until it has passed its acceptance tests. This means that new acceptance tests must be created for each iteration or the development team will report zero progress.	C215.4	**BTL2**
12	**What are the objectives of testing?** i. Testing is a process of executing a program with the intend of finding an error. ii. A good test case is one that has high probability of finding an undiscovered error. iii. A successful test is one that uncovers as an-yet undiscovered error.	C215.4	**BTL5**
13	**What are the testing principles the software engineer must apply while performing the software testing? MAY/JUNE 2016, APRIL/MAY 2018** i. All tests should be traceable to customer requirements. ii. Tests should be planned long before testing begins.	C215.4	**BTL2**

	iii. The pareto principle can be applied to software testing-80% of all errors uncovered during testing will likely be traceable to 20% of all program modules. iv. Testing should begin “in the small” and progress toward testing “in the large”. v. Exhaustive testing is not possible. vi. To be most effective, an independent third party should conduct testing.		
14	**What are the two levels of testing?** i. Component testing Individual components are tested. Tests are derived from developer“s experience. ii. System Testing The group of components are integrated to create a system or sub- system is done.These tests are based on the system specification.	C215.4	**BTL5**
15	**What are the various testing activities?** i. Test planning ii. Test case design iii. Test execution iv. Data collection v. Effective evaluation	C215.4	**BTL5**
16	**What is equivalence partitioning?** Equivalence partitioning is a black box technique that divides the input domain into classes of data. From this data test cases can be derived. Equivalence class represents a set of valid or invalid states for input conditions.	C215.4	**BTL2**
17	**What methods are used for breaking very long expression and statements? <u>NOV/DEC2016</u>** Refactoring is done to break long expression and ststements.	C215.4	**BTL5**
16	**What are the various testing strategies for conventional software?** i. Unit testing ii. Integration testing. iii. Validation testing. iv. System testing.	C215.4	**BTL5**

18	**How can refactoring be made more effective? APR/MAY 2016** Refactoring improves nonfunctional attributes of the software. Advantages include improved code readability and reduced complexity; these can improve source-codemaintainability and create a more expressive internal architecture or object model to improve extensibility	C215.4	**BTL5**
19	**How will you test a simple loop NOV/DEC 2015** • A simple loop is tested in the following way: • Skip the entire loop. • Make 1 pass through the loop. • Make 2 passes through the loop. • Make x passes through the loop where x<y, n is the maximum number of passes through the loop. • Make "y","y-1","y+1" passes through the loop where "y" is the maximum number of allowable passes through the loop.	C215.4	**BTL2**
20	**What are the conditions exists after performing validation testing?** After performing the validation testing there exists two conditions. • The function or performance characteristics are according to the specifications and are accepted. • The requirement specifications are derived and the deficiency list is created. The deficiencies then can be resolved by establishing the proper communication with the customer.	C215.4	**BTL5**
21	**Distinguish between alpha and beta testing. MAY/JUNE 2016** • Alpha and beta testing are the types of acceptance testing. • Alpha test: The alpha testing is attesting in which the version of complete software is tested by the customer under the supervision of developer. This testing is performed at developer's site.	C215.4	**BTL6**

<table>
<tr><td></td><td>

- Beta test: The beta testing is a testing in which the version of the software is tested by the customer without the developer being present. This testing is performed at customer's site.

</td><td></td><td></td></tr>
<tr><td>22</td><td>

What are the various types of system testing?

1. Recovery testing – is intended to check the system‟ s ability to recover from failures.

2. Security testing – verifies that system protection mechanism prevent improper

penetration or data alteration.

3. Stress testing – Determines breakpoint of a system to establish maximum service level.

4. Performance testing – evaluates the run time performance of the software, especially real-time software.

</td><td>C215.4</td><td><b>BTL5</b></td></tr>
<tr><td>23</td><td>

Define debugging and **What are the common approaches in debugging?**

Debugging is defined as the process of removal of defect. It occurs as a consequence of successful testing

Brute force method: The memory dumps and run-time tracks are examined and program with

write statements is loaded to obtain clues to error causes.

Back tracking method: The source code is examined by looking backwards from symptom to

potential causes of errors.

Cause elimination method: This method uses binary partitioning to reduce the number of locations where errors can exists.

</td><td>C215.4</td><td><b>BTL6</b></td></tr>
<tr><td>24</td><td>

Distinguish between verification and validation. <u>NOV/DEC2016, NOV/DEC 2017, APRIL/MAY 2018</u>

<table>
<tr><th>Verification</th><th>Validation</th></tr>
<tr><td>Evaluates the intermediary products</td><td>Evaluates the final product to</td></tr>
</table>

</td><td>C215.4</td><td><b>BTL5</b></td></tr>
</table>

to check whether it meets the specific requirements of the particular phase	check whether it meets the business needs.
Checks whether the product is built as per the specified requirement and design specification.	It determines whether the software is fit for use and satisfy the business need.
Checks “Are we building the product right”?	Checks “Are we building the right product”?
This is done without executing the software	Is done with executing the software
Involves all the static testing techniques	Includes all the dynamic testing techniques.
Examples includes reviews, inspection and walkthrough	Example includes all types of testing like smoke, regression, functional, systems and UAT

25	**What is meant by structural testing?** In structural testing derivation of test cases is according to program structure. Hence knowledge of the program is used to identify additional test cases.	C215.4	**BTL5**
26	**What is the need for regression testing? <u>APR/MAY 2015</u>** The purpose of regression testing is to confirm that a recent program or code change has not adversely affected existing features. Regression testing is nothing but full or partial selection of already executed test cases which are re-executed to ensure existing functionalities work fine.	C215.4	**BTL5**
27	**Write about drivers and stubs**. **NOV/DEC 2017** Drivers and stub software need to be developed to test incompatible software.	C215.4	**BTL6**

	The "driver" is a program that accepts the test data and prints the relevant results. The "stub" is a subprogram that uses the module interfaces and performs the minimal data manipulation if required.		
28	**What is cyclomatic complexity?** Cyclomatic complexity is software metric that gives the quantitative Measure of logical complexity of the program.	C215.4	**BTL5**
29	**How to compute the cyclomatic complexity?** The cyclomatic complexity can be computed by any one of the following ways. 1. The numbers of regions of the flow graph correspond to the cyclomatic complexity. 2. Cyclomatic complexity (G), for the flow graph G, is defined as: V(G)=E-N+2, E -- number of flow graph edges, N -- number of flow graph nodes 3. V(G) = P+1 Where P is the number of predicate nodes contained in the flow graph.	C215.4	**BTL5**
30	**List out the applications of GUI? <u>April /May 2015</u>** GUI-Graphical User Interface- is a type of interface that allows users to interact with electronic devices through graphical icons and visual indicators such as secondary notation, as opposed to text-based interfaces, typed command labels or text navigation In addition to computers, GUIs can be found in hand-held devices such as MP3 players, portable media players, gaming devices and smaller household,smartphones office and industry equipment. Eg:Ticket booking, Inventory tool, Billing Machine, Windows OS	C215.4	**BTL5**
31	**What is flow graph notation and how it is important. <u>April /May 2015</u>** A control flow graph (CFG) in	C215.4	**BTL5**

	computer science is a representation, Using graph notation, of all paths that might be traversed through aprogram during its execution.		
32	**What is smoke testing ? APRIL /MAY 2017** **Smoke Testing**, also known as "Build Verification **Testing**", is a type of software **testing** that comprises of a non-exhaustive set of **tests** that aim at ensuring that the most important functions work. The results of this **testing** is used to decide if a build is stable enough to proceed with further **testing**.	C215.4	**BTL5**
33	**List testing strategies that address verification. Which types of testing address validation ? APRIL/MAY 2017** Verification involves all the static testing techniques. Examples includes reviews, inspection and walkthrough Validation includes all the dynamic testing techniques. Example includes all types of testing like smoke, regression, functional, systems and UAT	C215.4	**BTL5**
33	**What are the types of static testing tools?** There are three types of static testing tools. ➢ **Code based testing tools** : These tools take source code as input and generate test cases. ➢ **Specialized testing tools :** Using this language the detailed test specification can be written for each test case. ➢ **Requirement-based testing tools**: These tools help in designing the as per user requirements.	C215.4	**BTL6**
34	**What is done in test design step?** The details of the layout, tooling and standards required for test	C215.4	**BTL5**

	development are designed in this stage.		
35	**Distinguish between verification and validation**? Verification refers to the set of activities that ensure that software correctly implements a specific function. Validation refers to a different set of activities that ensure that the software that has been built is traceable to the customer requirements.	C215.4	**BTL5**
36	**Write about drivers and stubs?** Drivers and stub software need to be developed to test incompatible software. The “driver” is a program that accepts the test data and prints the relevant results. The “stub” is a subprogram that uses the module interfaces and performs the minimal data manipulation if required.	C215.4	**BTL5**
37	**Define debugging**. Debugging is defined as the process of removal of defect. It occurs as a consequence of successful testing.	C215.4	**BTL5**
38	**Define the terms:** **a) Graph Matrices.** **b) Connection Matrices.** Graph Matrices: • To develop software tool the data structure used is graph Matrix. • Square Matrix • Size equals number of nodes on the Flow graph **Connection Matrices:** • It Link Weight = 1= > Connection Exists • It Link Weight=1=>Connection Does not Exists.	C215.4	**BTL3**

39	**What errors are commonly found during Unit Testing?** Errors commonly found during Unit Testing are: • Misunderstood or incorrect arithmetic precedence • Mixed Mode Operations • Incorrect Initializations • Precision Accuracy • Incorrect Symbolic representation of expression.	C215.4	**BTL5**
40	**What problems may be encountered when Top-Down Integration is chosen?** Following problems may be encountered when Top Down Integration is chosen: • Develop stubs that perform limited functions that simulate the actual module. Integrate the software from the bottom of the hierarchy upward	C215.4	**BTL5**
41	**What are the Steps in Bottom-Up Integration?** Steps in Bottom-Up Integration are: • Low level components are combined into clusters perform specific software sub function. • Driver is written to coordinate test case input and output. • Cluster is tested.	C215.4	**BTL5**

42	**What is Flow Graph Notation?** Flow Graph Notation means Simple notation for representing Control Flow. It is drawn only when Logical Structure of component is complex.	C215.4	**BTL6**
43	**What is acceptance testing** **Acceptance testing :**This type of testing involves testing of the system with customer data if the system behaves as per customer need then it is accepted.	C215.4	**BTL5**
44	**What are the various testing strategies for conventional software?** The various testing strategies are: (i) Unit testing (ii) Integration testing (iii) Validation testing (iv) System testing.	C215.4	**BTL6**
45	**List some of the testing done during SDLC.** White box testing, black box testing, integration testing, system testing, installation testing. Regression testing, Acceptance testing.	C215.4	**BTL5**
46	**What is functionality testing?** It is a black box testing which exercises the basic functionality of the product from an external; perspective.	C215.4	**BTL5**

47	What are the steps carried out in installation testing? **Ans.** The steps carried out in installation testing are: • Packaging • Documenting • Installing • Verifying	C215.4	**BTL5**
48	**What are the objective of Formal Technical Reviews.** **Ans.** The Objective of Formal Technical Reviews are: • Uncover errors in function, logic and implementation for representation of software. • Software represented according to predefined standard. • Verify software under review meets requirements • Achieve software developed in Uniform Manner. • Make projects more manageable.	C215.4	**BTL6**
49	**Explain Integrated testing team model?** **Ans.** There in one project manage who manages both the development and the testing functions	C215.4	**BTL5**
50	**What are the common approaches in debugging?** **Ans.** The common approaches tin debugging are: • **Brute force method:** The memory dumps and run- time tracks are examined and program with write statements in loaded to obtain clues to error causes. • **Back tracking method:** The source code is examined by looking	C215.4	**BTL6**

	backwards from symptom to potential causes or errors. • **Causes eliminations method:** This method uses binary partitioning to reduce the number of location where errors can exists.		

PART –B

S.NO	QUESTIONS	CO	BLOOM'S LEVEL
1	**What is black box & white-box testing? Explain how basis path testing helps to derive test cases to test every statement of a program.NOV/DEC-12, APRIL/MAY 2015, NOV/DEC 2017, APRIL/MAY 2017** *Press-Pg-no- 424*	C215.4	**BTL5**
2	**Define: Regression testing. Distinguish: top-down and bottom-up integration. How is testing different from debugging? JustifyNOV/DEC-10, APRIL/MAY 2018** *Press-Pg-no- 394 , 411*	C215.4	**BTL5**
3	**Write a note on equivalence partitioning & boundary value analysis of black box testingAPR/MAY-16 , NOV/DEC-15** *Press-Pg-no- 434*	C215.4	**BTL6**
4	**What is unit testing? Why is it important? Explain the unit test consideration and test procedure.APR/MAY-11,MAY/JUN-13 NOV/DEC2015** *Press-Pg-no- 394*	C215.4	**BTL5**
5	**Explain Integration & debugging activities?MAY/JUN-15** *Press-Pg-no-411*	C215.4	**BTL6**
6	**Explain software testing types?APR/MAY-16, NOV/DEC 2015** *Press-Pg-no- 384*	C215.4	**BTL5**

7	**Write elaborately on unit testing and regression testing. How do you develop test suites.<u>APRIL/MAY-15, APRIL/MAY 2018</u>** *Press-Pg-no- 376*	C215.4	**BTL5**
8	**i.What is cyclomatic complexity? How to compute cyclomatic complexity <u>APRIL/MAY-15, NOV/DEC 2017</u>** *Press-Pg-no- 421*	C215.4	**BTL6**
9	**Explain integration testing in detail.<u>MAY/JUN-13, APRIL/MAY 2017, APRIL/MAY 2018</u>** *Press-Pg-no- 397*	C215.4	**BTL5**
10	**What is black box testing? Explain the different types of black box testing strategies with example?<u>NOV/DEC 2016</u>** *Press-Pg-no- 424*	C215.4	**BTL6**
11	**1. (a) Consider the pseudo code for simple subtraction given below: <u>NOV/DEC 2016, APRIL/MAY 2018</u>** **(1) program 'simple subtraction'** **(2) input (x,y)** **(3) output (x)** **(4) output (y)** **(5) if x> y then DO** **(6) x-y = z** **(7) else y –x = z** **(8) endif** **(9) output (z)** **(10) output "end program"** **Perform basis path testing and generate test cases.** **(b) What is refactoring? When is it needed? Explain with ex?** Refer class notes.	C215.4	**BTL5**
12	**Explain in detail about system testing**	C215.4	**BTL5**

	Pressman Pg no. 352- 358		
13	**Explain about the software testing strategies** **Pressman Pg no. 304- 312**	C215.4	**BTL5**
14	**Discuss in detail about test strategies for conventional software(May/June 2011)** Refer class notes	C215.4	**BTL5**
15	**Explain in detail about basic path testing.(May/Jun 2014)** **Pressman Pg no. 356- 362**	C215.4	**BTL6**

UNIT – 5

PART –A

S.NO	QUESTIONS	CO	BLOOM'S LEVEL
1	**What are the processes of risk management? NOV/DEC-10, NOV/DEC-12, NOV/DEC 2013,NOV/DEC2015** Risk identification Risk projection (estimation) Risk mitigation, monitoring, and management	C215.5	**BTL6**
2	**State the need for software configuration review. NOV/DEC-11** The intent of the review is to ensure that all elements of the software configuration have been properly developed, cataloged & have necessary detail to bolster the supportpfase of the software lifecycle.	C215.5	**BTL6**
3	**List any five CASE tools classified by function in the taxonomy of CASE tools NOV/DEC-11** 1. project planning tools 2. metrics & management tools	C215.5	**BTL6**

	3. prototyping tools 4. Re- engineering tools 5. documentation tools.		
4	**Define error, fault and failure. NOV/DEC-10** Error – it is a state that can lead to a system behavior that is unexpected by the System user. Fault- it is a characteristic of a software system that can lead to system error. Failure – it is an event that occurs at some point in time when the system does not Deliver a service as per user's expectation.	C215.5	**BTL5**
5	**What is project planning? APR/MAY-12, APR/MAY-15** The various types of plan is developed to support main software project plan which is concerned with schedule & budget. Types of project plan Quality plan, Validation plan, Configuration mgmt plan, Maintenance plan, Staff development plan.	C215.5	**BTL5**
6	**List the various types of software errors? APR/MAY-11, NOV/DEC-12** Reports detailing bugs in a program are commonly known as bug reports, defect reports, fault reports, problem reports, trouble reports, change requests.	C215.5	**BTL6**
7	**Differentiatebetween size oriented and function oriented metrics? MAY/JUN-13 MAY/JUNE 2016,NOV/DEC 2015** Size oriented metrics – it considers the size of the software that has been produced. The software organization maintains simple records in tabular form. Table entries are LOC, effort, defects, and project name. Function oriented metrics – it measures the functionality delivered by software. Function point based on software information domain and complexity.	C215.5	**BTL6**
8	**Define measure.(APRIL/MAY-2008)**	C215.5	**BTL5**

	Measure is defined as a quantitative indication of the extent, amount, dimension, or size of some attribute of a product or process.		
9	**How is productivity and cost related to function points? NOV/DEC2016** Software Productivity = Function Points / Inputs (persons/mnth) Cost = $ / Function Points (FP)	C215.5	**BTL6**
10	**What are the types of metrics? MAY/JUNE 2016** Direct metrics – It refers to immediately measurable attributes. Example – Lines of code,execution speed. Indirect metrics – It refers to the aspects that are not immediately quantifiable or measurable. Example – functionality of a program.	C215.5	**BTL6**
11	**What are the advantages and disadvantages of size measure?** **Advantages:** • Artifact of software development which is easily counted. • Many existing methods use LOC as a key input. • A large body of literature and data based on LOC already exists **Disadvantages:** This method is dependent upon the programming language. ■ This method is well designed but shorter program may get suffered. ■ It does not accommodate non procedural languages. ■ In early stage of development it is difficult to estimate LOC.	C215.5	**BTL6**
12	**Write short note on the various estimation techniques.** □Algorithmic cost modeling – the cost estimation is based on the size of the software. □Expert judgement – The experts from software development and the application domain use their exoerience to predict software	C215.5	**BTL6**

	costs. □Estimation by analogy – The cost of a project is computed by comparing the project to a similar project in the same application domain and then cost can be computed. □Parkinson's law – The cost is determined by available resources rather than by objective assessment. □Pricing to win – The project costs whatever the customer ready to spend it.		
13	**What is COCOMO model?** COnstructiveCOstMOdel is a cost model, which gives the estimate of number of man- months it will take to develop the software product.	C215.5	**BTL6**
14	**Give the procedure of the Delphi method.** 1. The co-ordinator presents a specification and estimation form to each expert. 2. Co-ordinator calls a group meeting in which the experts discuss estimation issues with the coordinator and each other. 3. Experts fill out forms anonymously. 4. Co-ordinator prepares and distributes a summary of the estimates. 5. The Co-ordinator then calls a group meeting.In this meeting the experts mainly discuss the points where their estimates vary widely. 6. The experts again fill out forms anonymously. 7. Again co-ordinator edits and summarizes the forms,repeating steps5 and 6 until the co-ordinator is satisfied with the overallprediction synthesized from experts.	C215.5	**BTL6**
15	**What are the metrics computed during error tracking activity?** Errors per requirement specification page. Errors per component-design level Errors per component-code level DRE-requirement analysis DRE-architectural analysis DRE-component level design	C215.5	**BTL5**

	DRE-coding.		
16	**What is risk management? <u>NOV/DEC2016</u>** **Risk management** is the identification, assessment, and prioritization of risks followed by coordinated and economical application of resources to minimize, monitor, and control the probability and/or impact of unfortunate eventsor to maximize the realization of opportunities. Risk management's objective is to assure uncertainty does not deflect the endeavor from the business goals.	C215.5	**BTL6**
17	**What is software maintenance?** Software maintenance is an activity in which program is modified after it has been put into use.	C215.5	**BTL6**
18	**Will exhaustive testing guarantee that the program is 100% correct? <u>APR/MAY 2016</u>** No, even exhaustive testing will not guarantee that the program is 100 percent correct. There are too many variables to consider.	C215.5	**BTL6**
19	**What are the types of software maintenance?** • Corrective maintenance – Means the maintenance for correcting the software faults. • Adaptive maintenance – Means maintenance for adapting the change in environment. • Perfective maintenance – Means modifying or enhancing the system to meet the new requirements. • Preventive maintenance – Means changes made to improve future maintainability	C215.5	**BTL6**
20	**How the CASE tools are classified?** CASE tools can be classified by a. By function or use b. By user type(e.g. manager,tester),or c. By stage in software engineering process (e.g.requirements,test).	C215.5	**BTL6**

21	**Dinguish between direct & indirect measures of metrics.** Direct metrics is directly measurable attribute(lines of code execution speed,size of memory. Indirect metrics: these are the aspects that are not immediately measurable.(functionality,reliabblity,maintainability)	C215.5	**BTL6**
22	**List down few process and product metrics. MAY/JUNE 2016** 1.size metrics-It is used for measuring the size of the software.(local based metric,FP based metric) 2.complexity metric- A software module can be described by a control flow graph.(cyclomatic complexity, McCabe complexity) 3.quality metric- (Defects,reliabilitymetric,Maintainability)	C215.5	**BTL6**
23	**Define software measure.** It is a numeric value for a attribute of a software product or process. Types: 1.Direct measure 2.indirect measure	C215.5	**BTL6**
24	**List out the different approaches to size of the software**. 1.LOC-computing the line of code 2.FP-computing function point of the program.	C215.5	**BTL6**
25	**An organic software occupies 15000 LOC.how many programmers are needed to complete?(NOV/DEC-12)** System=organic Lines of coding=15k LOC $E=a_b(KLOC)b_b$ $=2.4(15)^{1.05}$ =41 persons per month $D=c_b(e)d_b$ $=2.5(41)^{0.38}$ =10 months P=41/10 P=4 persons.	C215.5	**BTL6**

	4 persons are needed.		
26	**What is error tracking?(APRIL/MAY-14)** It is a process of finding out and correcting the errors that may occur during the software development process at various stages such as software design,coding or documenting.	C215.5	**BTL6**
27	**What are the types of static testing tools?** There are three types of static testing tools. • Code based testing tools – These tools take source code as input and generate test cases. • Specialized testing tools – Using this language the detailed test specification can be written for each test case. • Requirement-based testing tools – These tools help in designing the test cases as per user requirements.	C215.5	**BTL6**
28	**What are the productivity measures and list its type. APRIL/MAY 2017** Productivity is an overall measure of the ability to produce a good or service. More specifically, productivity is the measure of how specified resources are managed to accomplish timely objectives as stated in terms of quantity and quality. Productivity may also be defined as an index that measures output (goods and services) relative to the input (labor, materials, energy, etc., used to produce the output). there are two major ways to increase productivity: increase the numerator (output) or decrease the denominator (input).	C215.5	**BTL6**
29	**Define ZIPF's law.** The probability of occurrence of words or other items starts high and tapers off. Thus, a few occur very often while many others occur rarely. Formal Definition: Pn ~ 1/na, where Pn is the frequency of occurrence of the nth ranked item and a is close to 1.	C215.5	**BTL6**
30	**List out the principles of project scheduling. NOV/DEC2017** Software project scheduling is an activity that distributes estimated effort	C215.5	**BTL6**

	across the planed project duration by allocating the effort to specific software engineering tasks. First, a macroscopic schedule is developed. a detailed schedule is redefined for each entry in the macroscopic schedule. A schedule evolves over time. Basic principles guide software project scheduling: - Compartmentalization - Interdependency - Time allocation - Effort allocation - Effort validation - Defined responsibilities - Defined outcomes - Defined milestones		
31	**Write a note on Risk information sheet. NOV/DEC 2017** A risk information sheet is a means of capturing information about a risk. Risk information sheets are used to document new risks as they are identified. They are also used to modify information as risks are managed. It is a form that can be submitted to the appropriate person or included in a database with other project risks. In the absence of a database, this becomes a primary means of documenting and retaining information about a risk.	C215.5	**BTL6**
32	**List two customer related and technology related risks. APRIL/MAY 2017**	C215.5	**BTL6**

	customer related risks •Customer relationship management may be fragmented. •New methods with which to improve customer service and reduce related costs are not utilized. •Lack of knowledge on the part of one section of an enterprise regarding interactions with a customer on the part of another can lead to customer frustration and embarrassment. •Inability to respond to market demands caused by lack of integration among order-entry systems or, even worse, due to infrastructure. • Lack of visibility of the order status along the whole supply chain. **Technology related risk :** • Architecture risk • Artificial intelligence risk • Audit risk • Availability		
33	**What is EVA ? APRIL/MAY 2018** Earned Value Analysis (**EVA**) is an industry standard method of measuring a project's progress at any given point in time, forecasting its completion date and final cost, and analyzing variances in the schedule and budget as the project proceeds.	C215.5	**BTL6**
34	**Identify The Types Of Maintenance for each of the following<u>APRIL/MAY 2018</u>** **Correcting the Software Faults .** **Adapting the change in environment** There are four **types of maintenance**, namely, **corrective**, adaptive, perfective, and **preventive**. ... **Correctivemaintenance** dealswiththe **repair** of **faults** or **defects** found in day-today system functions. ... In the event of a system **failure** due to an error, actions are taken to restore the operation of the **software** system.	C215.5	**BTL6**

35	**What is cost schedule?** Cost schedule shows the planned cumulative expenditure cost by the use of resource overtime	C215.5	**BTL6**
36	**What is RMMM?** **Ans.** RMMM stands for Risk Mitigation, Monitoring and Management Plan. It is also called Risk Aversion.	C215.5	**BTL6**
37	**What Is Risk mitigation?** **Ans.** Mitigation is a possible means if minimizing or even avoiding the Impact of risk.	C215.5	**BTL6**
38	**What are the factors that lead to Risk?** **Ans.** The factors that lead to Risk are: • Estimation errors. • Planning assumptions. • Business risks.	C215.5	**BTL6**
39	**What are the test points?** Test points allow data to be inspected or modified at various points in the system	C215.5	**BTL6**
40	**What is refactoring?** A small change to a database schema which improves its design	C215.5	**BTL6**
41	**Explain the common risk tools and techniques.** **Ans.** There are at least six different ways of identifying the potential risks. These are:	C215.5	**BTL6**

	• Examining organizational history • Preparing checklists • Information buying • Framework based risk categorization • Simulation • Decision trees.		
42	**What is called support risk?** **Ans.** Support risk is the degree of uncertainty fiat the resultant software will be easy to correct, adapt and enhance	C215.5	**BTL6**
43	**What Is Risk?** **Ans.** Risks are events that are usually beyond the planner's control.	C215.5	**BTL6**
44	**What are the Dimensions of Risk quantification?** **Ans.** Probability and the impact of Risk.	C215.5	**BTL6**
45	**What is meant by Delphi method?** The Delphi technique is an estimation technique intended to active a common agreement for estimation efforts.	C215.5	**BTL6**
46	**What is meant by CASE tools?** The computer aided software engineering tools automatic the project	C215.5	**BTL6**

	management activities, manage all the work products. The CASE tools assist to perform various activities such as analysis, design, coding and testing.		
47	**What are the three phases of Risk management?** **Ans.** The three phases of risk management are: Risk identification, Risk Quantification, and Risk mitigation.	C215.5	**BTL6**
48	**What are the factors that lead to Risk?** **Ans.** The factors that lead to Risk are: • Estimation errors. • Planning assumptions. • Business risks.	C215.5	**BTL6**
49	**What is meant by software project scheduling?** Software project scheduling is an activity that distributes estimated effort across the planned project duration by allocating the effort to specified software engineering tasks.	C215.5	**BTL6**
50	**What are the various steps under risk analysis?** **Ans.** The various steps under risk analysis are: • Risk Estimation. • Risk identification. • Risk evaluation.	C215.5	**BTL6**

PART –B

S.NO	QUESTIONS	CO	BLOOM'S LEVEL
1	**(a) Elaborate on the series of tasks of a software configuration management process.** **(b)Describe function point analysis with a neat exampleNOV/DEC 2013** *Press-Pg-no- 771, 685*	C215.5	**BTL6**
2	**Explain make/buy decision & discuss Putnam resource allocation model & derive time & effort equation?APRIL/MAY2016** *Somm – Pg-no- 634 , Press-Pg-no- 726*	C215.5	**BTL6**
3	**Explain the various CASE tools for project management and how they are useful in achieving the objectivesAPRIL/MAY-15** *Press-Pg-no- 645*	C215.5	**BTL6**
4	**Brief about calculating Earned value measuresAPR/MAY-12,APRIL/MAY 2018**	C215.5	**BTL5**

	Press-Pg-no- 722		
5	**Define Risk. Explain the needs and activities or risk management?<u>APR/MAY-15 , NOV/DEC2015 ,NOV/DEC 2017</u>** *Press-Pg-no- 726*	C215.5	**BTL6**
6	**Explain about all COCOMO models?<u>NOV/DEC 2015, APRIL/MAY2016, APRIL/MAY 2017, APRIL/MAY 2018</u>** *Press-Pg-no- 691*	C215.5	**BTL5**
7	**Write about software maintenance, PERT - CPM for scheduling , RMMP <u>NOV/DEC-12</u>** *Somm – Pg-no- 514, Press-Pg-no- 716, 739*	C215.5	**BTL6**
8	**Describe steps involved in project scheduling process, project timeline chart and task network. <u>MAY/JUN-15, APRIL/MAY 2018</u>** *Press-Pg-no- 708*	C215.5	**BTL6**
9	**(a) Suppose you have a budgeted cost of a project as Rs. 9,00,000. The project is to be completed in 9 months. After a month you have completed 10 percent of project at a total expense of Rs. 1,00,000. The planned completion should have been 15 percent. You need to determine whether the project is on-time and on budget? Use Earned value analysis approach and interpret<u>NOV/DEC 2016</u>** **(b) Consider the following function point components and their complexity. If the total degree of influence is 52, find the estimated function points.** **Function type Estimated count complexity** **FED 2 7** **GHD 4** **10** **HJI 22 4** **BU 16 5**	C215.5	**BTL5**

	BJ 24 4 *Refer class notes*		
10	**Describe in detail COCOMO model for software cost estimation. Use it to estimate the effort required to build software for a simple ATM that produce 12 screens, 10 reports and has 80 software components. Assume average complexity and average developer maturity. Use application composition model with object points.<u>NOV/DEC 2016, NOV/DEC 2017</u>** *Refer class notes*	C215.5	**BTL6**
11	**Explain the process of function point analysis?explain function point analysis with sample cases for componentfor different complexity <u>APRIL/MAY 2018</u>** Refer class notes	C215.5	**BTL6**
12	**Discuss on the various software cost estimation techniques. (April/MayApr/May 2008)** Refer class notes	C215.5	**BTL6**
13	**Explain the process of Delphi method ? advantages and disadvantages (Nov/Dec 2013)** Refer class notes	C215.5	**BTL5**
14	**Explain about Risk management (May/Jun 2014)** **Som Pg.no. 324-336**	C215.5	**BTL6**
15	**Give detail explanation about Scheduling and Tracking** Refer class notes	C215.5	**BTL6**