


CMR INSTITUTE OF TECHNOLOGY											
	USN										

Internal Assessment Test – III, January 2020

Sub:	SOFTWARE ENGINEERING							Code:	18MCA14
Date:	14.01.2020	Duration:	90 mins	Max Marks:	50	Sem:	I	Branch:	MCA

Answer **ONE FULL QUESTION** from each part


Marks	OBE	
	CO	RBT

Part-I

1	a) What is the relationship between an architecture and system-level design?	05	CO5	L1
	b) What is the difference between a flow chart and a structure chart?	05	CO5	L1
	(OR)			
2	Discuss some approaches on how you can use metrics to guide you in design to produce a design that is easy to modify.	10	CO5	L2

Part-II

3	Explain in detail about the issues and challenges related to the distributed system design.	10	CO5	L2
	(OR)			
4	Explain in detail about different architectural patterns of distributed systems.	10	CO5	L3

CMR INSTITUTE OF TECHNOLOGY											
	USN										

Internal Assessment Test – III, January 2020

Sub:	SOFTWARE ENGINEERING							Code:	18MCA14
Date:	14.01.2020	Duration:	90 mins	Max Marks:	50	Sem:	I	Branch:	MCA

Answer **ONE FULL QUESTION** from each part

Marks	OBE	
	CO	RBT

Part-I

1	a) What is the relationship between an architecture and system-level design?	05	CO5	L1
	b) What is the difference between a flow chart and a structure chart?	05	CO5	L1
	(OR)			
2	Discuss some approaches on how you can use metrics to guide you in design to produce a design that is easy to modify.	10	CO5	L2

Part-II

3	Explain in detail about the issues and challenges related to the distributed system design.	10	CO5	L2
	(OR)			
4	Explain in detail about different architectural patterns of distributed systems.	10	CO5	L3

CMR INSTITUTE OF TECHNOLOGY
Department of Computer Applications-PG Studies
Third Internal Assessment Test- Scheme and Solution
18MCA14-Software Engineering

Questions Number	Expected Answer	Marks Break up	Maximum Marks
1a	<p>What is the relationship between an architecture and system-level design?</p> <ul style="list-style-type: none"> • Design: UML diagram/flow chart/simple wireframes (for UI) for a specific module/part of the system. • Architecture: component diagram (showing how The different module of the system communicates with each other and other systems), what language is to be used, patterns...? • Software design is about designing the individual modules / components. What are the responsibilities, functions, of module x? Of class Y? What can it do, and what not? What design patterns can be used? • So in short, Software architecture is more about the design of the entire system, while software design emphasizes on module / component / class level. • Architecture is the bigger picture: the choice of frameworks, languages, scope, goals, and high-level methodologies (Rational, waterfall, agile, etc.). • Design is the smaller picture: the plan for how code will be organized; how the contracts between different parts of the system will look; the ongoing <i>implementation</i> of the project's methodologies and goals. Specifications are written during this stage. 	5	5
1b	<p>What is the difference between a flow chart and a structure chart?</p> <ul style="list-style-type: none"> • A structure chart differs from a flow chart in three principal ways • Flow chart is easy to understand for beginners. Structure chart is difficult for beginners to understand as solution is displayed in hierarchical order, like a tree. • Flowchart is to explain processes • A structure chart highlights the relationship between modules. Only rectangles are used in the diagrammatic representation of a structure chart. • Data interchange among different modules is not represented in a flow chart. • Sequential ordering of tasks inherent in a flow chart is suppressed in a structure chart 	5	5

CMR INSTITUTE OF TECHNOLOGY
Department of Computer Applications-PG Studies
Third Internal Assessment Test- Scheme and Solution
18MCA14-Software Engineering

2	<p>Discuss some approaches on how you can use metrics to guide you in design to produce a design that is easy to modify.</p> <p>Main objective:</p> <p>Does the design implement the requirements Analysis for performance, efficiency, etc may also be done if formal languages used for design representation, tools can help Design reviews remain the most common approach for verification.</p> <ul style="list-style-type: none"> ■ Basic purpose to provide a quantitative evaluation of the design (so the final product can be better) ■ Size is always a metric – after design it can be more accurately estimated <ul style="list-style-type: none"> • Number of modules and estimated size of each is one approach ■ Complexity is another metric of interest – will discuss a few metrics <p>Network Metrics: Stability Metrics Information Flow Metrics</p>	10	10
3	<p>Explain in detail about the issues and challenges related to the distributed system design.</p> <ul style="list-style-type: none"> • Resource sharing (hardware and software) • Openness (standard protocols allow equipment and software from different vendors to be combined) • Concurrency (parallel processing to enhance performance) • Scalability (increased throughput by adding new resources up to capacity of network) • Fault tolerance (potential to continue in operation after a fault has occurred) 	10	10
4	<p>Explain in detail about different architectural patterns of distributed systems.</p> <ul style="list-style-type: none"> ■ Master-slave: used in real-time systems for which guaranteed interaction response times are required. <ul style="list-style-type: none"> ○ Commonly used in real-time systems with separate processors associated with <u>data acquisition</u>, <u>data processing</u>, and <u>computation and actuator management</u>. ○ A “Master” process is usually responsible for computation, coordination, and communications; it controls the “slave” processes. 	10	10

CMR INSTITUTE OF TECHNOLOGY
Department of Computer Applications-PG Studies
Third Internal Assessment Test- Scheme and Solution
18MCA14-Software Engineering

	<ul style="list-style-type: none"> ○ “Slave” processes are dedicated to specific actions, e.g., the acquisition of data from an array of sensors. 		
5	<p>Explain in detail about software as a service.</p> <p>Software as a service (SaaS) is a software distribution model in which a third-party provider hosts applications and makes them available to customers over the Internet. SaaS is one of three main categories of <u>cloud computing</u>.</p> <p>SaaS is typically accessed by users using a thin client via a web browser. SaaS has become a common delivery model for many business applications, including office software, messaging software, payroll processing software, DBMS software, management software, CAD software, gamification, virtualization,[3] accounting, collaboration, customer relationship management (CRM), Management Information Systems (MIS), enterprise resource planning (ERP), invoicing, human resource management (HRM), talent acquisition, learning management systems, content management (CM), Geographic Information Systems (GIS), and service desk management. SaaS has been incorporated into the strategy of nearly all leading software companies.</p>	10	10
6	<p>Describe in detail about Constructive Cost Model for effort estimation of planning a software project development</p> <p>The software development effort estimation is an essential activity before any software project initiation. We will illustrate how to easily estimate the software effort using known estimation techniques which are Function Points Analysis (FPA) and Constructive Cost Model (COCOMO).</p> <p>Estimation Process</p> <ol style="list-style-type: none"> 1- Scoping 2- Decomposition 3- Sizing 4- Expert and Peer Review 5- Estimation Finalization 	10	10

CMR INSTITUTE OF TECHNOLOGY
Department of Computer Applications-PG Studies
Third Internal Assessment Test- Scheme and Solution
18MCA14-Software Engineering

7	<p>Discuss in detail about project scheduling and staffing.</p> <p>Project Scheduling in a project refers to roadmap of all activities to be done with specified order and within time slot allotted to each activity. Project managers tend to define various tasks, and project milestones and them arrange them keeping various factors in mind. They look for tasks lie in critical path in the schedule, which are necessary to complete in specific manner (because of task interdependency) and strictly within the time allocated. Arrangement of tasks which lies out of critical path are less likely to impact over all schedule of the project.</p> <p>For scheduling a project, it is necessary to -</p> <ul style="list-style-type: none"> • Break down the project tasks into smaller, manageable form • Find out various tasks and correlate them • Estimate time frame required for each task • Divide time into work-units • Assign adequate number of work-units for each task • Calculate total time required for the project from start to finish 	10	10
8	<p>Explain the steps involved in the risk management process in detail</p> <p>Risk is an expectation of loss, a potential problem that may or may not occur in the future. It is generally caused due to lack of information, control or time. A possibility of suffering from loss in software development process is called a software risk. Loss can be anything, increase in production cost, development of poor quality software, not being able to complete the project on time. Software risk exists because the future is uncertain and there are many known and unknown things that cannot be incorporated in the project plan. A software risk can be of two types (a) internal risks that are within the control of the project manager and (2) external risks that are beyond the control of project manager. Risk management is carried out to:</p> <ol style="list-style-type: none"> 1. Identify the risk 2. Reduce the impact of risk 3. Reduce the probability or likelihood of risk 4. Risk monitoring 	10	10

CMR INSTITUTE OF TECHNOLOGY
Department of Computer Applications-PG Studies
Third Internal Assessment Test- Scheme and Solution
18MCA14-Software Engineering

9	<p>Write in detail about white box testing techniques with suitable example</p> <p>White box testing strategy deals with the internal logic and structure of the code. White box testing is also called as glass, structural, open box or clear box testing. The tests written based on the white box testing strategy incorporate coverage of the code written, branches, paths, statements and internal logic of the code etc.</p> <p>In order to implement white box testing, the tester has to deal with the code and hence is needed to possess knowledge of coding and logic i.e. internal working of the code. White box test also needs the tester to look into the code and find out which unit/statement/chunk of the code is malfunctioning.</p>	10	10
10	<p>Write in detail about any two black box testing techniques with example</p> <p>Black box testing, which is also known as behavioral, opaque-box, closed-box, specification-based or eye-to-eye testing, is a Software Testing method that analyses the functionality of a software/application without knowing much about the internal structure/design of the item that is being tested and compares the input value with the output value.</p> <p>The main focus in black box testing is on the functionality of the system as a whole. The term 'behavioral testing' is also used for black box testing. Behavioral test design is slightly different from the black-box test design because the use of internal knowledge isn't strictly forbidden, but it's still discouraged.</p>	10	10