GMR Institute of Technology
An Autonomous Institute Affiliated to JNTUK, Kakinada

GMRIT Training Tomorrow's Engineers Today

Continuous Assessment Question Paper

GMRIT/IA/F-5
REV: 03

U.G.	**CSE**				Degree	Bachelor of Technology	
Academic Year	2021 – 2022	Sem.	6	Test	2	Date of Exam	/05/2022
Course Code	**19CS603**			Course Title	**Software Engineering**		

SCHEME

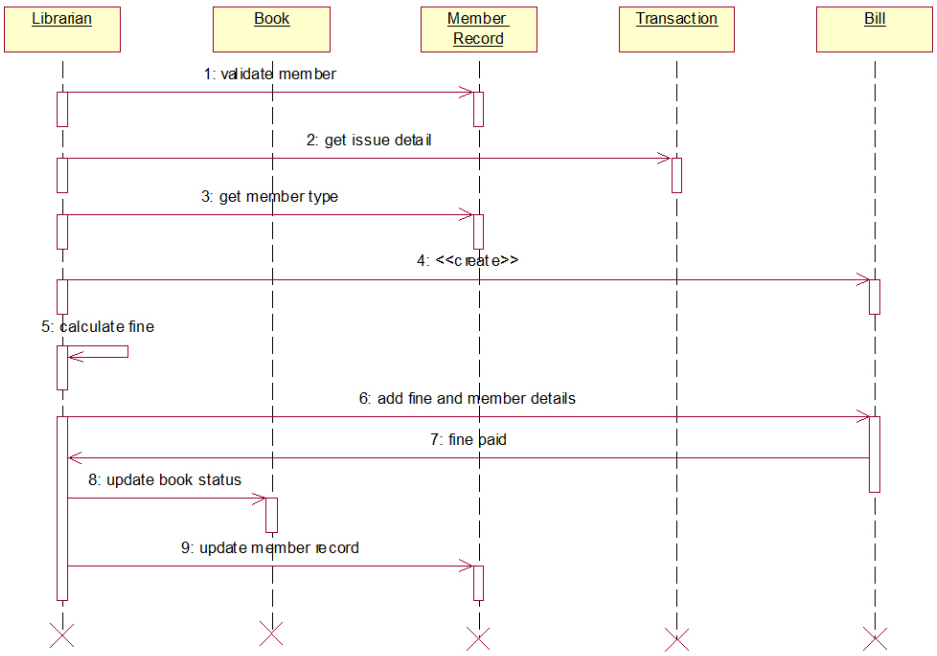
PART A (5 X 2= 10 Marks) (Answer all the questions)

No.	Questions (1 to 5)
1	Data design is the first design activity, which results in less complex, modular and efficient program structure.
2	➢ Class diagram. ➢ Object diagram. ➢ Component diagram. ➢ Deployment diagram.
3	➢ Top down approach ➢ Bottom Up approach
4	They are applied in research, development, production, process control, quality control, testing, management, etc.
5	➢ **Product metrics** − Describes the characteristics of the product such as size, complexity, design features, performance, and quality level. ➢ **Process metrics** − These characteristics can be used to improve the development and maintenance activities of the software. ➢ **Project metrics** − This metrics describe the project characteristics and execution. Examples include the number of software developers, the staffing pattern over the life cycle of the software, cost, schedule, and productivity ➢ Reliability ➢ Performance ➢ Usability ➢ Correctness ➢ Maintainability ➢ Integrity ➢ Security

PART B (3 X 10= 30 Marks) Answer ALL questions

No	Questions (6 to 11)	Marks
6	"Coupling" describes the relationships between modules, and "cohesion" describes the relationships within them. A reduction in interconnectedness between modules (or classes) is therefore achieved via a reduction in coupling. On the other hand, well-designed modules (or classes) should have some purpose; all the elements should be associated with a single task. This means that in a good design, the elements within a module (or class) should have internal cohesion.	10

	✓ Unit test can be done within a single module ✓ Cohesion of a single module/component is the degree to which its responsibilities form a meaningful unit; higher cohesion is better. ✓ Someone had vague reference to decomposability here. Clarification? ✓ How about: 'Cohesion is inversely proportional to the number of responsibilities a module/component has.' ✓ Coupling between modules/components is their degree of mutual interdependence; lower coupling is better. size: number of connections between routines intimacy: the directness of the connection between routines visibility: the prominence of the connection between routines flexibility: the ease of changing the connections between routines ✓ A first-order principle of software architecture is to increase cohesion and reduce coupling. ✓ so having coupling in program is not correct ✓ reduce coupling in your program it will favour Unit testing	
7	• It is also known as Behavioral Testing, is a software testing method in which the internal structure/design/implementation of the item being tested is not known to the tester. • These tests can be functional or non-functional, though usually functional. **Black Box Testing Techniques** • ***Equivalence Partitioning:*** It is a software test design technique that involves dividing input values into valid and invalid partitions and selecting representative values from each partition as test data. • ***Boundary Value Analysis:*** It is a software test design technique that involves the determination of boundaries for input values and selecting values that are at the boundaries and just inside/ outside of the boundaries as test data. • ***Cause-Effect Graphing:*** It is a software test design technique that involves identifying the cases (input conditions) and effects (output conditions), producing a Cause-Effect Graph, and generating test cases accordingly.	10
8	• A software project can be concerned with a large variety of risks. • In order to be adept to systematically identify the significant risks which might affect a software project, it is essential to classify risks into different classes. • The project manager can then check which risks from each class are relevant to the project. • There are three main classifications of risks which can affect a software project: **Project risks** **Technical risks**	10

	Business risks **Strategies for Risk mitigation** Following are the steps to manage risks effectively in an organization: • **Risk Identification** • **Risk Quantification** • **Risk Response** • **Risk Monitoring and Control**	
✓	✓ Data-centered architecture ✓ Data-flow architecture ✓ Call and return architectures ✓ Layered architectures ✓ Model-View-Controller(MVC) ✓ Repository Architecture ✓ client–server architecture ✓ Pipe and Filter	10
10	 Write All the Notations of Sequence Diagram.	10
11	**Black box testing techniques** • Black Box Testing Technique • Boundary Value Analysis (BVA) • Equivalence Class Partitioning **Apply any of above two Black box Testing Techniques.**	10