|  |
| --- |
| **KONGU ENGINEERING COLLEGE, PERUNDURAI - 638 060** |
| **SEMESTER ODD |CONTINUOUS ASSESSMENT TEST – II** |
| (Regulations **2020**) |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Month and Year | : | October 2023 | Roll Number | : |  |
| Programme | : | B.Tech. | Date | : | 09-10-2023 |
| Branch | : | IT | Time | : | 09.15am - 10.45am |
| Semester | : | V | Duration | : | 1½ Hours |
| Course Code | : | 20ITT53 | Max. Marks | : | 50 |
| Course Name | : | Software Engineering |  |  |  |

**Answer Key**

|  |  |  |  |
| --- | --- | --- | --- |
|  | Compare static and dynamic model in UML.   * Static model: It represents the structural or static aspect of the system. They focus on elements that exist at a specific point in time and do not consider the behavior or interactions between these elements. * Dynamic model: It is viewed as a collection of procedures or behaviors that reflect the behavior of the system over time. They capture the runtime aspects of a system, including the sequence of activities, events, and state transitions. | | |
|  | What is the significance of the system boundary in a use case diagram?  A rectangle shape representing the boundary between the actors and the system. It separates the use cases that are internal to a system from the actors that are external to the system. | | |
|  | Apply UML Use Case Diagrams to model the interactions in a library management system. Include at least three actors.  Diagram similar to the following is accepted,  Use case diagram of the simplified library management system. | Download  Scientific Diagram | | |
|  | List the essential components of a sequence diagram, and how are they represented?  Objects – rectangle shape  Objects lifeline – dashed line drawn below each object  Messages – arrow between the lifelines of two objects  Activation bars - horizontal bars along the lifelines | | |
|  | Discuss the role of generalization in UML class diagrams and provide an example.  Generalization is used to model inheritance. The role of generalization in UML class diagrams is to establish an "is-a" relationship.  Example similar to the following is accepted, | | |
|  | How do you represent concurrent actions in an activity diagram?  In an Activity Diagram, concurrent actions or parallelism can be represented using two primary notations: fork and join nodes. | | |
|  | Differentiate between Deployment diagram and Component diagram in UML.  **Deployment diagram:** Visualize and specify the hardware components and infrastructure on which a software system is deployed or executed. They show the physical distribution of software components onto hardware nodes.  **Component diagram:** It represents the high-level architecture and structure of a software system by decomposing it into its major software components/modules and their relationships. | | |
|  | Outline the architectural design elements.  Architectural design elements give us an overall view of the software. The architectural design element is usually depicted as a set of interconnected subsystems. Each subsystem may have its own architecture. | | |
|  | Define refactoring.  Refactoring is the process of changing a software system in such a way that it does not alter the external behavior of the code [design] yet improves its internal structure. | | |
|  | What is an archetype? Give example.  An archetype is an abstraction (similar to a class) that represents one element of system behaviour. | | |
|  |  | | |
| 11. | i) | Draw a Collaboration diagram for a banking system, showcasing the user registration and new account creation process.  Diagram similar to the following is accepted, | (5) |
|  | ii) | Create a UML sequence diagram that illustrates the interaction between a Customer, Cart and Product. Include key lifelines, messages, and activations.  Diagram similar to the following is accepted,  C:\Users\Guna\Desktop\Untitled-2 copy.jpg | (5) |
| 12. | i) | Design a UML class diagram for a simple e-commerce system that includes classes for customers, products, and orders. Show the associations and multiplicity between these classes.  Diagram similar to the following is accepted, | (10) |
| 13. | i) | Develop an activity diagram depicting the process of withdrawing cash from an ATM, taking into account multiple scenarios, such as checking the balance, verifying the PIN, and executing the cash withdrawal. Provide explanations for each step and decision point in the diagram.  Diagram similar to the following is accepted, | (10) |
| 14. | i) | Write short notes on the design model.   * Data design elements   + Data model --> data structures   + Data model --> database architecture * Architectural design elements   + Application domain   + Analysis classes, their relationships, collaborations and behaviors are transformed into design realizations   + Patterns and “styles” * Interface design elements   + the user interface (UI)   + external interfaces to other systems, devices, networks or other producers or consumers of information   + internal interfaces between various design components. * Component design elements: describes the internal detail of each software component. * Deployment design elements: indicate how software functionality and subsystems will be allocated within the physical computing environment that will support the software. | (5) |
|  | ii) | Summarize Data-Centered and Layered architecture within the context of architectural styles.  Data-Centered Architecture:   * A data store (e.g., a fi le or database) resides at the center of this architecture and is accessed frequently by other components that update, add, delete, or otherwise modify data within the store. * Data-centered architectures promote integrability.     Layered Architecture:   * A number of different layers are defined, each accomplishing operations that progressively become closer to the machine instruction set. * At the outer layer, components service user interface operations. * At the inner layer, components perform operating system interfacing. * Intermediate layers provide utility services and application software functions. | (5) |