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

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Month and Year | : | November 2023 | Roll Number | : |  |
| Programme | : | B.Tech. | Date | : | 17-11-2023 |
| Branch | : | IT | Time | : | 02.30pm - 04.00pm |
| Semester | : | V | Duration | : | 1½ Hours |
| Course Code | : | 20ITT53 | Max. Marks | : | 50 |
| Course Name | : | Software Engineering |  |  |  |

**Answer Key**

|  |  |  |  |
| --- | --- | --- | --- |
|  | Illustrate on data model in user interface design.  A data model represents the structure and organization of the data that the interface interacts with. It defines how data is stored, retrieved, and manipulated within the system. A well-designed data model is crucial for creating a user interface that is intuitive, efficient, and meets the needs of the users. | | |
|  | Provide the syntax for defining an attribute in the component-level design.  name: type-expression 5 initial-value {property string}  where name is the attribute name, type expression is the data type, initial value is the value that the attribute takes when an object is created, and property-string defines a property or characteristic of the attribute. | | |
|  | Differentiate on Traditional and process-level view of component.  Traditional View:   * designed from scratch * a component is a functional element of a program that incorporates processing logic, the internal data structures that are required to implement the processing logic. * the component hierarchy is mapped into a module hierarchy.   Process-Related View:   * make use of existing software components or design patterns. * choose components or design patterns from the catalog and use them to populate the architecture. * these components have been created with reusability in mind | | |
|  | Mention the 3 golden rules of Interface design.   * Place the user in control * Reduce the user’s memory load. * Make the interface consistent. | | |
|  | Compare top-down and bottom-up integration testing approaches  Top-Down Integration Testing:   * Starts testing from the topmost module (main module or high-level modules) and gradually integrates lower-level modules. * Early identification of high-level issues. * Stub (temporary replacement for lower-level modules) creation may be complex.   Bottom-up integration testing:   * Begins testing with the lowest-level modules, gradually integrating higher-level modules. * Early identification of low-level issues. * Need to use driver programs to simulate higher-level modules. | | |
|  | List the type of system testing  Recovery, Security, Stress, Performance and Deployment | | |
|  | Define Debugging.  Debugging is the process of identifying, isolating, and fixing errors, defects, or bugs in a software program. The goal of debugging is to locate the root cause of unexpected behavior or issues in the code and then make the necessary corrections to ensure the program functions as intended. | | |
|  | Outline the steps involved in alpha testing.  Planning, Selection of Alpha Testers, Environment Setup, Distribution of Software, Test Execution, Feedback Collection, Regression Testing, Documentation Updates, Conclusion and Reporting. | | |
|  | Provide an example of a test case for a login feature of a website, including the test inputs and expected outcomes.  Answer similar to the following is acceptable,  Test Inputs:   * Username: ValidUsername * Password: ValidPassword   Steps to Execute:   * Navigate to the login page of the website. * Enter the valid username into the "Username" field. * Enter the valid password into the "Password" field. * Click on the "Login" button.   Expected Outcome:   * The user should be successfully logged in. | | |
|  | What is SCI?  Software Configuration Items are the individual components of a software system that are managed and controlled as part of the configuration management process. They are essential for maintaining the integrity, reliability, and traceability of a software project throughout its development and maintenance lifecycle. | | |
|  |  | | |
|  | i) | Given a scenario of developing an online bookstore system, identify at least three distinct components, outline their responsibilities, and explain how they interact to achieve the system's functionality.  Answer similar to the following is acceptable,  **User Authentication Component:**   * Manages user authentication and authorization. * Interacts with the User Interface (UI) for login requests, communicates with the Database Component to validate user credentials, and returns authentication status to the UI.   **Inventory Management Component:**   * Ensures accurate stock information and availability status. * Receives requests from the User Interface for book searches or updates, communicates with the Database Component to retrieve or modify book data, and provides search results or updates to the UI.   **Shopping Cart and Checkout Component:**   * Manages the order creation, calculates totals, and ensures a smooth checkout process. * Interacts with the User Interface to update the shopping cart, communicates with the Inventory Management Component to verify book availability, and communicates with the Database Component to store order information. | (10) |
|  | i) | Explain the concept of system testing in detail. Identify at least three common performance-related issues.   * System testing is a level of software testing where the entire integrated system is tested to evaluate its compliance with specified requirements. * It involves testing the system as a whole, including its interactions with external systems or interfaces. * The primary goal is to ensure that the software system behaves according to the defined functional and non-functional requirements. * System testing is typically performed after integration testing and before acceptance testing. * Any Performance-Related Issues similar to the following:   + Response Time Degradation   + Concurrency and Scalability Issues   + Memory Leaks | (5) |
|  | ii) | Write short notes on Software Configuration Management.  Software Configuration Management (SCM):   * SCM is a systematic approach to managing the development, change, and version control of software artifacts. * It involves identifying, controlling, and tracking changes to software configurations, ensuring consistency and traceability. * SCM includes version control, configuration identification, change management, and release management, facilitating collaboration and maintaining the integrity of software throughout its lifecycle. | (5) |
|  | i) | Your team is preparing for beta testing of a mobile application. Outline the steps you would take to select beta testers, distribute the application, and collect feedback. Explain how you would prioritize and address reported issues.  Answer similar to the following is acceptable,  Selecting Beta Testers:   * Identify a diverse group of potential users, including those with different devices, operating systems, and usage patterns.   Distributing the Application:   * Develop a streamlined process for distributing the beta version, leveraging app distribution platforms or beta testing services. * Provide clear instructions on how to install and use the beta version, including any specific testing instructions or areas of focus.   Collecting Feedback:   * Establish communication channels for beta testers to report issues, such as a dedicated feedback form, email, or a discussion forum.   Prioritizing and Addressing Issues:   * Categorize reported issues based on severity, impact on user experience, and frequency. * Prioritize critical issues that affect functionality, security, or user data. * Create a roadmap for addressing identified issues, considering both critical fixes and enhancements based on user feedback. | (10) |
|  | i) | What is regression testing, and why is it important in software development? Explain how it helps in maintaining software quality.   * Regression testing is the re-execution of some subset of tests that have already been conducted to ensure that changes have not propagated unintended side effects. * Regression testing helps to ensure that changes do not introduce unintended behavior or additional errors. * Regression testing may be conducted manually, by reexecuting a subset of all * test cases or using automated capture/playback tools. * Regression Testing Maintains Software Quality   + Ensures Stability   + Prevents Bug Proliferation   + Validates Code Changes   + Sustains User Experience | (10) |