School of Computer Science and Engineering

Winter Semester 2023-24

Continuous Assessment Test – II

SLOT: C1 + TC1

Programme Name & Branch: B. Tech. Computer Science & Engineering (Core and Specializations)

Course Name & Code: Software Engineering & BCSE301L

Class Number (s):

VL2023240500597, VL2023240500632, VL2023240500656, VL2023240500723, VL2023240500697,
VL2023240500700, VL2023240500594, VL2023240500622, VL2023240500714, VL2023240500588,
VL2023240500707, VL2023240500590, VL2023240500609, VL2023240500648, VL2023240500730,
VL2023240500585, VL2023240500603, VL2023240500650, VL2023240500665, VL2023240500614,
VL2023240500673, VL2023240500719, VL2023240500736

Exam Duration: 90 Min   **Maximum Marks: 50**

Sl. No:	Questions	Marks	CO	BL

1.	Consider the scenario of students' academic record management software. In this a set of courses are created. Each course consists of a unique course number, number of credits, and the syllabus. Students are admitted to courses. Each student's details include his roll number, address, semester number and the courses registered for the semester. The marks of student for various subjects he credited are keyed in. Once the marks are keyed in, the semester weighted average (SWA) is calculated. The recent marks of the student are added to his previous marks and a weighted average based on the credit points for various subjects is calculated. The marks for the current semester are formatted and printed. The SWA appears on the report. A check must be made to determine if a student should be placed on the Vice-Chancellor's list for academic excellence. This is determined based on whether a student scores an SWA of 85 or higher. If the SWA is lower than 50, the student is placed on a conditional standing.(Conditional standing is the academic status assigned to students identified as not maintaining satisfactory academic performance in their course and who will be required to improve their academic performance and meet certain conditions if they wish to continue in their course).	10	3	4

a) Create an Entity Relationship diagram that captures the information on the scenario. (3 Marks)
b) Design a context diagram and DFD level 1 diagram for the scenario.

(3+ 4 Marks)

//Main keywords of the diagram listed here.

Entities:

1. Course (Course_Number, Number_of_Credits, Syllabus)
2. Student (Roll_Number, Address, Semester_Number)
3. Subject (Subject_Name, Credit_Points, Marks)
4. Semester_Registration (Student_Roll_Number, Course_Number)
5. Transcript (Student_Roll_Number, Semester_Number, SWA)

Relationships:

1. Student enrolls in Course (Many-to-Many)
2. Student has Marks for Subject (One-to-Many)
3. Student has Transcript for Semester (One-to-One)

Attributes:

- Course_Number (Primary Key for Course)
- Number_of_Credits
- Syllabus
- Roll_Number (Primary Key for Student)
- Address
- Semester_Number
- Subject_Name
- Credit_Points
- Marks
- SWA

This ER diagram represents the relationships between courses, students, subjects, and transcripts in the academic record management software. Each student enrolls in courses, has marks for different subjects, and has a transcript for each semester which includes the SWA. The system calculates the SWA

based on the marks and determines if a student should be on the Vice-Chancellor's or placed on conditional standing.

Level 0 DFD:

- Processes:
    1. Manage Courses
    2. Admit Students
    3. Key in Student Marks
    4. Calculate SWA
    5. Calculate Weighted Average
    6. Format and Print Marks
    7. Check for VC list placement
    8. Check for conditional standing

Level 1 DFD:

- Manage Courses:
    - o Input: Course details
    - o Output: Course database
    - o Process: Create course
- Admit Students:
    - o Input: Student details and selected courses
    - o Output: Student database
    - o Process: Admission process
- Key in Student Marks:
    - o Input: Student marks
    - o Output: Updated student database with marks
    - o Process: Input marks
- Calculate SWA:
    - o Input: Student marks
    - o Output: SWA
    - o Process: Calculate SWA
- Calculate Weighted Average:
    - o Input: Previous marks, current semester marks
    - o Output: Weighted average
    - o Process: Calculate weighted average
- Format and Print Marks:
    - o Input: Student marks
    - o Output: Printed marks
    - o Process: Format and print marks
- Check for VC list placement:
    - o Input: SWA
    - o Output: VC list placement result
    - o Process: Check SWA for VC list
- Check for conditional standing:
    - o Input: SWA
    - o Output: Conditional standing result
    - o Process: Check SWA for conditional standing

This is a basic overview of the DFDs for the students' academic record management software. Keywords alone mentioned here..

2.	a) A simplified account system of **a university** is introduced as follows. Each student has a unique **University** Account which contains an account id and a student name. Based on this mother account, a student will have a Link Account, which has an email address and password. Students are allowed to reset the password, but cannot change the email address. The link account is required when accessing Blackboard platform and Zoom software. Particularly, an important part of Blackboard is a list of courses and it provides operations like adding new courses, downloading, and uploading homework. Similarly, a list of meetings is an important part of Zoom and it can create new meetings or delete existing ones. Moreover, each Course is identified by a course id and course name and it has at least one meeting and one homework. A Meeting has a unique meeting id and an invitation link. When joining a meeting by its id, the password is required. Finally, each Homework has its own course id, homework no., and a list of questions. **Draw a static structure diagram by identifying the classes, attributes and their relationship among the classes.** (6 marks)	10	3	4
	b) Draw a diagram to model the interaction among active objects within a system using life line for the specific action **-the process of joining the online lecture of the course BCSE301L**. (4 Marks) There are four objects involved: Student, Zoom application, Zoom server, and University account server –			
3.	a) Imagine you are developing firmware for a microcontroller-based IoT device that monitors environmental conditions such as temperature, humidity, and air quality in a smart home. The device continuously samples sensor data and sends periodic updates to a central server for analysis and visualization. To efficiently handle asynchronous events and minimize CPU overhead, Identify the appropriate architecture control style for the scenario and justify the same. (5 Marks) Answer: Event driven – (Interrupt driven). (Expected to write the justification and advantage of this method) b) How does temporal cohesion contribute to the overall organization and efficiency of a software module's functionality? Provide an example from a real-world application where temporal cohesion is utilized effectively. (5 Marks)	10	3	4

	One way temporal cohesion contributes to the overall organization and efficiency of a software module's functionality is by making it easier to understand and maintain the code. When activities that need to occur at a specific point in time are grouped together, it becomes more clear to developers how they are related and can be executed in sequence. This reduces confusion and potential errors in the code, making it easier to debug and update in the future. An example of where temporal cohesion is utilized effectively is in an online shopping application. When a user adds an item to their shopping cart, various activities need to occur in a specific sequence, such as updating the total price, displaying a confirmation message, and storing the item in the user's cart. By grouping these activities together based on when they need to occur, the code becomes more organized and efficient. This makes it easier for developers to understand and maintain the functionality of the shopping cart feature, ensuring a smoother user experience for customers.			
4.	a) You are testing a new e-commerce website for a retail clothing store. The website allows users to browse and purchase clothing items online. Describe how you would approach testing the checkout process on the website to ensure a smooth and efficient user experience (4 Marks) 1. Understand the checkout process: -. This includes adding items to the cart, entering shipping and billing information, selecting payment method, reviewing the order, and completing the final purchase. 2. Test on different devices and browsers: Ensure that the checkout process works seamlessly across various devices and web browsers. Test on desktop computers, laptops, tablets, and smartphones to make sure the website is responsive and user-friendly. 3. Test different payment methods: Verify that all available payment methods (credit/debit cards, PayPal, Apple Pay, etc.) are functioning correctly and securely. Make test transactions to confirm that payments are processed accurately and without any errors. 4. Validate form fields and error messages: Test the validation of form fields such as email addresses, shipping addresses, and credit card information. Verify that appropriate error messages are displayed when incorrect or incomplete information is entered. 5. Test order confirmation and email notifications: After completing a purchase, confirm that the order is recorded correctly in the system and that the user receives a confirmation email with details of their purchase. Check for any discrepancies between the order placed and the order confirmation received. 6. Perform load testing: Simulate heavy traffic on the website to test how the checkout process handles a large number of simultaneous users. Ensure that the website remains responsive and does not experience any slowdowns or crashes during peak times. 7. Test security measures: Check for secure connections (HTTPS), SSL certificates, and payment security protocols to protect users' personal and financial information. Verify that sensitive data is encrypted and stored securely. 8. Seek feedback from real users: Consider conducting user testing sessions or collecting feedback from actual customers to identify any pain points or areas	10	4	**3**

for improvement in the checkout process. Use this feedback to make necessary adjustments and enhancements to the website

<mark>Or</mark> –Theory Aspects

1. The content model for the WebApp is reviewed to uncover errors.

2. The interface model is reviewed to ensure that all use cases can be accommodated.

3. The design model for the WebApp is reviewed to uncover navigation errors.

4. The user interface is tested to uncover errors in presentation and/or navigation mechanics.

5. Each functional component is unit tested.

6. Navigation throughout the architecture is tested.

7. The WebApp is implemented in a variety of different environmental configurations and is tested for compatibility with each configuration.

8. Security tests are conducted in an attempt to exploit vulnerabilities in theWebApp or within its environment.

9. Performance tests are conducted.

10. The WebApp is tested by a controlled and monitored population of end users.

The results of their interaction with the system are evaluated for errors.

b) You are assigned with testing a banking application's transfer functionality. The system allows users to transfer funds between their own accounts and to other accounts within the same bank. Perform Black box testing method to ensure the transfer functionality is working correctly in the bank. (Make your own assumptions.) (6 Marks)

- The system allows users to transfer funds within the same bank only, not to accounts in other banks.
- Users can transfer funds between their own accounts or to other accounts within the same bank.
- The transfer functionality includes entering the recipient's account number, the amount to transfer, and any optional message.
- There are two types of accounts in the bank: savings account and checking account.

Boundary Value Analysis:

1. Transfer amount:

- Lower boundary: The minimum transfer amount allowed is $1.
- Just above the minimum
- Upper boundary: The maximum transfer amount allowed is $10,000.
- Just below the maximum value

- Just above the maximum value

2. Account balances:

- Lower boundary: The minimum balance allowed to transfer funds is $0 (assuming users cannot transfer more funds than they have).
- Upper boundary: The maximum balance allowed in an account is $100,000.

- Account numbers:

    Check with the length of account number

Equivalence Class Testing:

1. Valid transfer scenarios:

- Transfer between own accounts (savings to checking, checking to savings)
- Transfer to another user's savings account
- Transfer to another user's checking account

2. Invalid transfer scenarios:

- Transfer amount is less than the minimum allowed
- Transfer amount is greater than the maximum allowed
- Insufficient funds in the sender's account
- Invalid recipient account number (e.g., incorrect length or format)

By performing boundary value analysis and equivalence class testing, we can ensure that the transfer functionality in the banking application works correctly under various scenarios and edge cases.

| 5. | Consider the following program segment

1.Void Selection Sort (int array1[]) {
2.   int size=array1.length;
3.   for (int i=0; i<size; i++){
4.   int minIndex=i;
5.   for (int j=i+1; j<size; j++) {
6.   if(array1[j] < array1[minIndex])
7.   minIndex=j; }
8.   int temp= array1[minIndex];
9.   array1[minIndex] =array1[i];
10.   array1[i] =temp;
11   .}
12. }
a) Prepare the CFG and determine the cyclomatic complexity of the selection sort function.
b) Design a test suite for the selection sort that satisfies the Path coverage (Show the important steps in your test suite design method).
(5+5 Marks) | 10 | 4 | 4 |

The number of regions in the CFG are:

- 3 --> 4 --> 5 --> 8 --> 9 --> 10 --> 3
- 5 --> 6 --> 5
- 5 --> 6 --> 7 --> 5
- Common External region
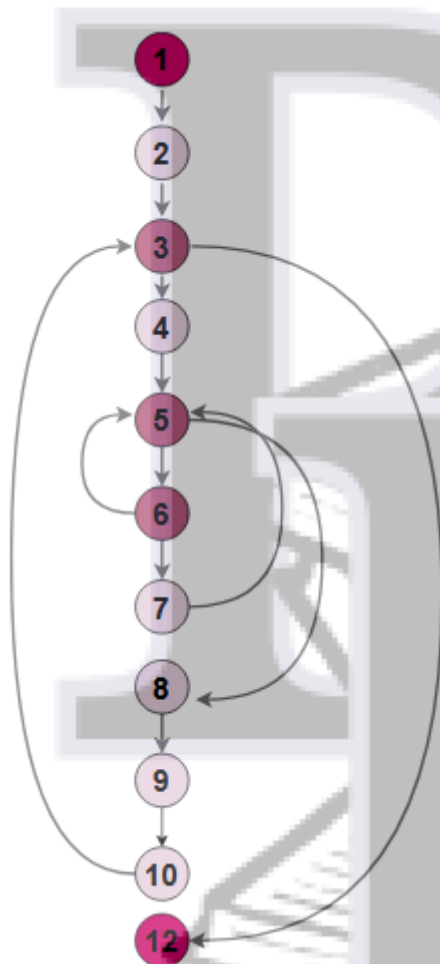
Cyclomatic complexity = 4

Test case suite -Need to check with the path coverage. [5]

Identifying the path and validating.