VIT
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)
CHENNAI

| Reg. Number: | 23MCA2030 |
|---|---|

# Continuous Assessment Test (CAT) – II - April 2024

| | | | | | |
|---|---|---|---|---|---|
| Programme | : | Master of Computer Applications | Semester | : | Winter 2023-24 |
| Course Code & Course Title | : | PMCA614L - Software Testing | Class Number | : | CH2023240501409 |
| Faculty | : | Dr. Renjith P N | Slot | : | G2 |
| Duration | : | 90 Minutes | Max. Mark | : | 50 |

**General Instructions:**

- Write only your registration number on the question paper in the box provided and do not write other information.

**Answer all questions**

| Q. No | Description | Marks |
|---|---|---|
| 1 | As the project manager overseeing the deployment of a College Management System (CMS) developed by your team, consider the following logical steps:<br><br>**a)** Identifying and elaborating on the major acceptance criteria crucial for evaluating the CMS. Justify the selection of these criteria based on the specific requirements and objectives of the college management system. **[4 Marks]**<br>**b)** Develop a comprehensive acceptance test plan. This plan should delineate the testing approach, methodologies, tools, and resources required to execute acceptance testing effectively. It serves as a blueprint for ensuring thorough evaluation of the CMS's functionality and performance. **[4 Marks]**<br>**c)** Discuss strategies for managing risks and mitigating potential challenges encountered during the testing process. **[2 Marks]** | 10 |
| 2 | **a)** Outline strategies for incorporating Path Testing into the broader testing strategy for the healthcare management system, including integration with other testing methodologies. **[4 Marks]**<br>**b)** Given the provided pseudocode and test set, perform the following tasks:<br><br>`1. main () {`<br>`2. int x, y, p;`<br>`3. input (x, y);`<br>`4. if (x < y)`<br>`5. p = g1(x, y);`<br>`6. else`<br>`7. p = g2(x, y);`<br>`8. endif`<br>`9. output (p);`<br>`10. end`<br>`11. }`<br><br>*g1:*<br>`1. int g1 (int a, b) {`<br>`2. int a, b;`<br>`3. if (a + 1 == b)`<br>`4. return (a*a);`<br>`5. else`<br>`6. return (b*b);`<br><br>*g2:*<br>`1. int g2 (int a, b) {`<br>`2. int a, b;`<br>`3. if (a == (b + 1))`<br>`4. return (b*b);`<br>`5. else`<br>`6. return (a*a);`<br><br>**c)** Consider the following test set:<br>t1: <x=1, y=3><br>t2: <x=2, y=1><br>t3: <x=1, y=2><br>i. Construct a Control Flow Graph to visualize the program's control flow. **[2 Marks]**<br>ii. Generate Execution Traces for each test case to illustrate the flow of execution through the program. **[2 Marks]**<br>iii. Create a syntax tree to represent the structure of the pseudocode. **[2 Marks]**<br><br>Ensure clarity and accuracy in your representations. | 10 |

| | | |
|---|---|---|
| 3 | **a)** Elucidate McCabe's Cyclomatic Complexity, considering its significance in software engineering and its implications for code quality and maintainability. **[2 marks]**<br>**b)** Perform comprehensive code analysis, including Control Flow Graph generation, Cyclomatic Complexity calculation, and Data-Flow Testing, for the provided Java program. **[8 Marks]**<br><br>1. public class ControlFlowGraph<br>2. {<br>3. public static void main(String[] args)<br>4. {<br>5. int x = 10;<br>6. int y = 20;<br>7. if (x < y)<br>8. {<br>9. System.out.println("x is less than y");<br>10. } else {<br>11. System.out.println("x is greater than or equal to y");<br>12. }<br>13. for (int i = 0; i < 5; i++)<br>14. {<br>15. if (i % 2 == 0)<br>16. {<br>17. System.out.println("Even: " + i);<br>18. }<br>19. else<br>20. {<br>21. System.out.println("Odd: " + i);<br>22. }<br>23. }<br>24. int i = 0;<br>25. while (i < 5) {<br>26. System.out.println("While loop: " + i);<br>27. i++;<br>28. }<br>29. do {<br>30. System.out.println("Do-while loop: " + i);<br>31. i--;<br>32. } while (i > 0);<br>33.<br>34. switch (x) {<br>35. case 5:<br>36. System.out.println("x is 5");<br>37. break;<br>38. case 10:<br>39. System.out.println("x is 10");<br>40. break;<br>41. default:<br>42. System.out.println("x is neither 5 nor 10");<br>43. }<br>44. }<br>45. } | 10 |
| 4 | **a)** You're a software developer responsible for testing the ZipCode web service, offering ZtoC and ZtoA functions initially for US zip codes. Modifications include ZtoC requiring country input and a new ZtoT service for time zones. Design a testing strategy covering scenarios, input validation, boundaries, and methods to ensure reliability across various usage scenarios. **[5 Marks]**<br><br>**b)** You are a software engineering intern at a technology consultancy firm, currently engaged in evaluating the performance of a major e-commerce company's online shopping platform. This platform has been experiencing frequent slowdowns and occasional outages, particularly during peak traffic times. Your internship project requires you to identify common bottlenecks in the application's performance and propose strategies to overcome them. **[5 Marks]** | 10 |
| 5 | Consider you are part of a development team working on a collaborative document editing platform that operates across multiple devices and platforms, such as web browsers, mobile apps, and desktop applications.<br>**a)** How would you leverage call graph-based integration testing to ensure seamless communication and synchronization among these diverse components? **[2 Marks]**<br>**b)** Outline specific scenarios where call graph analysis could uncover potential integration issues, such as data inconsistencies or synchronization delays, and propose strategies for mitigating these issues through targeted testing and optimization of communication protocols. **[5 Marks]**<br>**c)** Discuss how you would adapt your approach to handle the dynamic nature of user interactions and evolving feature sets in a collaborative editing environment. **[3 Marks]** | 10 |

**********All the best **********