School of Computer Science and Engineering
Winter Semester 2023-2024 Continuous Assessment Test – II

SLOT: B1+TB1

Programme Name & Branch: M.Tech. (Integrated) CSE & M.Tech. (Integrated) CSE with Specialization in Data Science

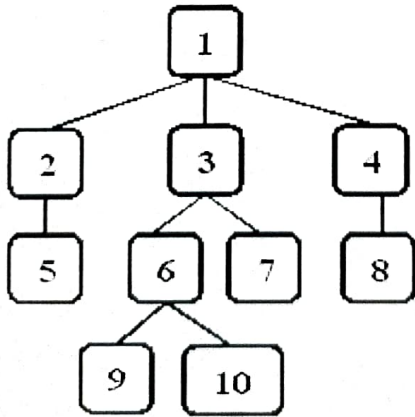**Course Name & code:** CSI3014 Software Verification and Validation

Class Number (s): VL2023240502425, VL2023240502509

Faculty Name (s): Prof. Narayanan Prasanth N, Prof. Jothi K R

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

General instruction(s): Answer all the questions

Q. No.	Question	Max Marks
1.	As a tester, you are assigned to prepare a Requirements Traceability Matrix (RTM) for a hotel reservation system. The RTM should ensure 100% requirement coverage with respect to the test cases. The two primary modules of the application are **Reservation** and **Payment**. Prepare three business and technical requirements for the given modules with atleast two test cases for each requirement.	10
2.	The modular design hierarchy of a flight booking application is shown below. Do the following tasks: a. How many **drivers and stubs** are required for the given decomposition tree? (2) b. Show how testing can be performed using **depth-first integration and bottom-up integration techniques.** (3+3) c. Calculate the **integration testing effort**. (2) 	10
3.	As a chairman of an inspection committee, you are assigned to inspect a Software Requirements Specification (SRS) document developed for an IPL auctioning system. Illustrate the process to be followed to inspect the SRS. In addition, prepare the inspection meeting notice, inspection defect list, and inspection defect summary.	10

4.	A program reads the data of the employees in a company by taking the following inputs and printing them: • Name of the employee (max. 15 valid characters, A-Z, a-z, space) • Employee ID (10 characters) • Designation (up to 20 characters) Design test cases for this program using Boundary Value Checking and Robustness Checking.	10
5.	Perform Basis Path Testing (BPT) for the code snippet given below. Do the following as a part of BPT: a) Draw the Control flow / DD graph for the code. b) Calculate the Cyclomatic Complexity using any two methods. c) List all the independent paths. d) Design test cases from the independent paths	10

```c
#include <stdio.h>
int main()
{
  char op;
  double first, second;
  printf("Enter an operator (+, -, *, /): ");
  scanf("%c", &op);
  printf("Enter two operands: ");
  scanf("%lf %lf", &first, &second);
  switch (op) {
    case '+':
    printf("%.1lf + %.1lf = %.1lf", first, second, first + second);
    break;
    case '-':
    printf("%.1lf - %.1lf = %.1lf", first, second, first - second);
    break;
    case '*':
    printf("%.1lf * %.1lf = %.1lf", first, second, first * second);
    break;
    case '/':
    printf("%.1lf / %.1lf = %.1lf", first, second, first / second);
    break;
    default:
    printf("Error! operator is not correct");
  }
  return 0;
}
```

ALL THE BEST