# END-SEMESTER EXAMINATION, May-2024
## Computer Science Workshop2 (CSE3141)

**Programme: B.Tech(CSE/CSIT)** **Semester: 4th**
**Full Marks: 60** **Time: 3 Hours**

| Subject/Course Learning Outcome | *Taxonomy Level | Ques. Nos. | Marks |
|---|---|---|---|
| Understanding Object-oriented programming, generic and collection classes, and applying them to solve different problems. | L1, L2, L3 | Q1 | 6 |
| Understanding Error handling, garbage collection, string, I/O operation, and file management of Java. And apply it to solve related problems. | L1, L2, L3 | Q2 | 6 |
| Learning different data structures and applying them to solve different problems and analyzing their effectiveness in different problem-solving. Understanding and applying Lambdas and Functional programming using Java. | L1, L2, L3 | Q3.Q4 Q5,Q6 | 12+12 |
| Understanding multithreading and reactive programming of Java, and applying it to solve related problems. | L1, L2, L3 | Q7,Q8 | 12 |
| Learning spring and spring MVC of Java and applying it to solve different problems. | L1, L2, L3 | Q9,Q10 | 12 |

*Bloom's taxonomy levels: Remembering (L1), Understanding (L2), Application (L3), Analysis (L4), Evaluation (L5), Creation (L6)

Answer all questions. Each question carries equal mark.

| | | | |
|---|---|---|---|
| 1. | (a) | Write a program to create **Complex** class, having member real and imaginary. Add the required constructor, set and get method. | 2 |
| | (b) | Create a class **ComplexApp** add a method that takes two complex object and find its summation. If any of the complex number is not in proper format it throws ComplexNumberFormat exception. | 2 |
| | (c) | Create a main function in **ComplexApp** for creation of the proper complex object handle the exception and call the methods for execution. | 2 |
| 2. | (a) | Create a class, define a static function in the class, which takes a mathematical expression as its arguments and return the | 2 |

| | | | |
|---|---|---|---|
| | | operator and operand present in the expression. | |
| | (b) | Define a static function in the above created class which takes the return result of the above function and a file name as its argument. The function writes the return result in that file. | 2 |
| | (c) | Create static function in the above created class which takes a file name and read the result and display it. Add a main function the created class to execute the above functions. | 2 |
| 3. | (a) | Create a class named **Graph** with members adj a two-dimensional matrix, n represent the number of nodes. Add the required method and constructor. | 2 |
| | (b) | Create a **GraphApp** class add a method ***BFS(Graph g, int s)*** to it which traverse the graph using breadth first search. | 2 |
| | (c) | Add a method ***DFS(Graph g)*** to **GraphApp** which traverses the graph using depth first search. | 2 |
| 4. | (a) | Add a method ***path(int s, int d)*** to **GraphApp** class which print the path between two node. | 2 |
| | (b) | Add a method ***printAdj(int node)*** to **GraphApp** class which prints the adjacent of a given node. | 2 |
| | (c) | Create a main method to invoke the above created method for execution.<br><br>*Note: Write one program for Q3 and Q4.* | 2 |
| 5. | (a) | Write a program to create an **Employee** class that has members' empid, name, date of joining and date of birth. Date of joining and date of birth both are **LocalDate** type. Add the required constructor. | 2 |
| | (b) | Add the required setter and getter methods. | 2 |
| | (c) | Add a method to the **Employee** class which prints the age and years of experience of the employee. | 2 |
| 6. | (a) | Create another class **EmpApp.** Add a method to the emp class which takes two Employee object and print the senior among them according to experience. | 2 |
| | (b) | Adda method to EmpApp which sort an array of Employee object according to empid. Use lambda function to override the compare/compareTo method. | 2 |
| | (c) | Call all the above created methods for execution.<br><br>*Note: Write one program for Q5 and Q6.* | 2 |
| 7. | (a) | Create a class named **MyArray** has member variable and int array and size of the array. Add the required method and set and get method to it. | 2 |
| | (b) | Add a synchronized method to **MyArray** which reads the number for the array. | 2 |
| | (c) | Add a method with synchronized block which prints the content of the array. | 2 |
| 8. | (a) | Create a Thread class that reads the elements for the array. | 2 |
| | (b) | Create another Thread class that prints the elements of the array. | 2 |
| | (c) | Create **ArrayApp** class add the required method to create two thread one is for reading the elements of the array and another is for printing the elements of the array.<br>*Note: Write one program for Q7 and Q8.* | 2 |
| 9. | (a) | Create a **Model** class having members modelNo, number of cylinder. Add the required constructor, set and get method. | 2 |
| | (b) | Create a **Car** class having members name, color and Model. Add the required constructor, set and get methods. | 2 |
| | (c) | Create an xml file for creation of the object using spring inverse of controle. | 2 |
| 10 | (a) | Create an xml file for dependency injection using property. | 2 |
| | (b) | Add lines to inject Model object which is the member of the Car class. | 2 |
| | (c) | Create a class **CarApp** to create the object of the Car class and print the details.<br>*Note: The object is created by the above created xml file.*<br>*Note: Write one program for Q9 and Q10.* | 2 |
| | | *End of Questions* | |