

MID-SEMESTER EXAMINATION, April-2024
Computer Science Workshop 2 (CSE 3141)

Programme: B.Tech. (CSE)
Full Marks: 30

Semester: 4th
Time: 2 Hours

Subject/Course Learning Outcome	*Taxonomy Level	Ques. Nos.	Marks
Understanding Object-oriented programming, generic and collection class and applying it to solve different problems.	L1, L2, L3	Q.1, Q.2	6+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	Q.3, Q.4, Q.5	6+6+6
Learning different data structure and applying it to solve different problems and analyze its effectiveness on different problem-solving. Understanding and applying Lambdas and Functional programming using Java.	L1,L2, L3		
Understanding multithreading and reactive programming of java, and applying it to solve related problems.	L1,L2,L3		
Learning spring and spring MVC of Java and applying it to solve different problems	L1,L2, L3		
Understanding and learning Hibernate and applying it to solve different problems.	L1,L2, L3		

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

Answer all questions. Each question carries equal mark.

- Define an interface named **Shape** with two abstract methods: `area()` and `perimeter()`. 2
 - Create two classes, **Circle** and **Rectangle**, both of which implement the **Shape** interface. In the Circle class, override the `area()` and `perimeter()` methods to print appropriate messages indicating the area and perimeter for a circle. Similarly, in the Rectangle class, override the same methods to display messages specific to a rectangle's area and perimeter. 2

- (c) Implement method overloading in both classes by providing multiple versions of the `area()` method, each accepting different parameters such as radius. Finally, create a Main class to instantiate objects of both **Circle** and **Rectangle** classes, test their overridden methods, and demonstrate method overloading by invoking different versions of the `area()` method. 2

Note: Write one program for Q1 a, b, c.

2. (a) Create a Class **Car** that has modelNo, name, and price as member variables. Add the required constructor and methods to it. 2

Note: Don't close the Car class keep some space to add some other method to it.

- (b) Override the required method, which is used for sorting, and searching car objects according to car price. 2
- (c) Create another class named **CarApp** and create a `carShowroom` which is an `ArrayList` of car objects. Sort the `carShowroom` according to price. Search for a particular car object according to price. Display the cars after sorting. 2

Note: Write one program for Q2 a, b, c.

3. (a) Write a Java program to create a class named **MyArithmeticException** by extending the required class. Add the required method and member variable to this class. 2
- (b) Create another class **CalculatorApp**. Add the main method to it which reads two integer numbers and performs addition, subtraction, multiplication, and division operations, and prints the results. 2
- (c) Add the required technique which throw and handles the **MyArithmeticException** exception if the first number is smaller than the second number. Also, add the required technique to handle the divide by zero exception. 2

Note: Write one program for Q3 a, b, c.

4. (a) Write a Java program to create an **Employee** class having name, age, and date of birth(dd/mm/yyyy). Add the required method and constructors to it. Add a method to this class that generates a password that takes the last name and year from the date of birth and returns it. 2

Example:

Input: If name is Ramesh Kumar and date of birth is

23/07/1993

Output: Password is Kumar1993

- (b) Add a method to the **Employee** class that takes a string that consists of the name, date of birth, and age as its argument create a student object from it and return it. 2

- (c) Add a method to the **Employee** class that prints the details of the student in the below format 2
- First name:
Middle name:
Last name:
DOB:dd-mm-yyyy
Age:

Invoke the above-created method for execution.

Note: Write one program for Q4 a, b, c.

5. (a) Write a Java program that creates an array of integers and reads some elements for the array. Create a static function that takes a number as an argument and checks whether the number is prime or not. 2
- (b) Create a static function that takes a file name and opens the file if it is present otherwise shows a proper message without terminating the program. If the file is present, then it stores all prime numbers present in the array in the file. 2
- (c) Create a static function that takes a file name of the above-created file and opens the file if it is present otherwise shows the proper message without terminating the program. Display the prime numbers which are present in the file. 2

Note: Write one program for Q5 a, b, c.

End of Questions