Department of ICT, MIT, Manipal

3rd Sem (IT/CCE) Object Oriented Programming (ICT 2122) FISAC

ABOUT THE SUBMISSION:

1. SHOULD BE HAND WRITTEN.

2. DO NOT WRITE THE QUESTION, WRITE ONLY THE SOLUTION.

3. USE A4 SHEETS(USE BOTH SIDES TO WRITE)

4. SPECIFY : YOUR NAME, REGISTRATION NUMBER, SECTION

5. DEADLINE : 31-10-2024

Q. No	Question
1	Create a java application which creates class Warehouse{String productId, int Quantity, int unitcost} class Order{int orderedQuanity, int OrderID, String date} class bookOrderThread{checkAvailableQuantity(), computeTotCost()} Create multiple threads for placing orders for the same productId. Ensure while generating the cost for the orderID, there is no conflict and is thread safe. Generate user defined exceptions for the following cases: 1. If Quantity is 0, “Order cannot be placed” 2. If orderedQuantity is 0, “Order quantity is invalid”, 3. For the case where ordered Qauntity is more than the available stock display “out of Stock for” ProductId. 4. Check if the placed order exist for the product Id else display “ProductID doesnot exist”. Once order is placed, display the sorted order details based on date.
2	Create an interface Func{ sort(int no), sort_date(String doj)}. Create below defined classes which uses appropriate sort method. Class Student{String name, int rank, String regNo, String Branchcode},

	class Faculty{String name, String DOI, String Dept, String empCode} a) Sort the students according to rank and then assign the regNo starting with code "2022" followed by "BranchCode" followed by counter value starting from 001 for every BranchCode. Reg no should be unique and sample looks like "2022CCE001" or "2022IT006". b) Sort the faculties according to DoJ. DoJ sample is of the format "dd-MMM-YYYY" or "dd-MM-YYYY". Throw user defined exception of DoJ if is not in the supposed format with message "Invalid date format". Assume all the faculties joined in 2002. Once sorted then assign the empCode in the format "MAHE" followed by "Dept" followed by counter value starting from 001 for every Dept. Sample looks like "MAHEICT001" or "MAHEECE005" c) Display the sorted faculties and students based on set(branchCode+regNo) and (Dept+ empcode) respectively with full details. d) Initialize appropriate constructors.
3	Write a java program to simulate the library information system by creating **a) Packages** :users (Package to handle user-related functionality), books (Package to handle book-related functionality), exceptions (Package to handle custom exceptions) b) **Interface:** To define an interface LibraryOperations to outline common operations like borrowBook, returnBook, calculate penalty for not returning c) **Classes**: User: Base class for user information (name, user ID). LibraryUser: Inherits from User and implements LibraryOperations. Book: Class for handling book details. d) Define custom exceptions like **BookNotAvailableException, BookNotBorrowedException, BookCannotBeIssued** to manage the exceptions. Implement it for multiple users.
4	Write a java program which will perform the following tasks: a) Thread1 creates a class containing employee information in the form of Employee class with EmpName, Empcode, Leave_type, DateFrom, DateTo and LeaveBalance as attributes for 5 employees. b) Thread2 is used to get the information of an employee who applies leave. If employee information is not found then raise an **exception EmployeeNotFound**, else it should be able to update the LeaveBalance based on the applied leave type. In the beginning of the academic year, the leaves are credited as below for every employee. No.of compensatory leaves - 13, No.of casual leaves - 12 for any other leaves -10. If an employee does not have the required no. of. leaves in his account when he applies, raise **an exception NoLeaves** and manage it. c) In main method, take input for at least 5 employees, display the details of all the updated leave information of only those employees who applied with leave type= "casual". Use synchronization to manage the discrepancy.

5	As a part of a larger application, the team leader of ABC enterprise has asked his intern to write a java program that takes as input Registration number as a **Number**. Intern should include routines in his program to handle issues such as data type mismatch , that would be seen during the program run. However the part of the code written by Intern is as follows and doesn't meet the requirement.
	// Input Registration Number
	boolean validRegistration = false;
	while (!validRegistration) {
	System.out.print("Enter Registration Number: ");
	registrationNumber = Integer.parseInt(scanner.nextLine());
	validRegistration = true;
	}
	Can you identify what issue would be seen during the program run and provide a solution to this ?

6	Implement a ThreadSafeQueue<T> class using generics that can store elements of any type. The class should provide the following functionalities: 1. enqueue(T item): Adds an item to the queue. If the queue is full (size > 10), it should throw a user defined QueueFullException. 2. dequeue(): Removes and returns an item from the queue. If the queue is empty, it should throw an user defined QueueEmptyException. 3. The enqueue and dequeue methods should be thread-safe using synchronized blocks to handle concurrent access.
7	Create a base class Task that has a method executeTask() which prints "Task is being executed". Then create a generic subclass SumTask<T> that extends Task and provides a thread-safe method sum(T a, T b) to sum two numbers. The method should throw a user defined NonNumericTypeException if the types are not numeric.
8	Develop a movie ticket booking application to book a movie ticket. Application should have following classes and methods. Class Movie having members: movieName, seat_type, and seatsAvailable with appropriate constructor to initialize movie object. Class BharathCenemasMultiplex with array of Movie objects and a **addMovie** method to add the movies to the BharathCenemasMultiplex, **listMovies** method to list all movies running in the multiplex along with the seats available and **bookTicket** method to book a ticket to the movie by specifying moviename, seat_type and number of tickets required. Add appropriate members based on requirements. After booking the tickets, tickets count should be updated. If the required number of tickets are not available then appropriate

	message need to be displayed. Also implement a method **listUser**, which list all users who booked ticket along with the movie name and ticket details.
9	A Seminar Hall has 15 seats and 20 users trying to book the ticket concurrently. Whenever you book a ticket for a seatnumber, it should be alloted on return if successful or should generate an user defined exception called SeatAlreadyBookedException. No two people should book the same seat and booking system works concurrently i.e all 20 users are trying to book the seat. Initially it should display all 15 seat numbers and finally it should display booking status. While booking each users name and aadhar number should be input. Name and seat number should come in status of booking
10	Define a generic class Box of size 3 to have capabilities to add an item, retrieve an item, check if the box is empty, and clear. Define a custom class Student. Demonstrate Box class with different type of objects such as Integer, String, and Student . Whenever the Box is full and you try to add an item it should generate a custom Exception called FullException. Also the process of adding and retrieving is done concurrently.