2/4 B.Tech. SECOND SEMESTER

CS4L1 **Object Oriented Programming (Java) Lab** **Credits: 4**

(Common to CSE, IT)

Required

Lecture: -- **Internal assessment: 25 marks**

Lab: 6 periods/week **Semester end examination: 50 marks**

Course context and Overview: This course is aimed at students who wish to learn how to develop applications in Java. This course will also provide an overview of Object Oriented Programming concepts using Java

---

Prerequisites: Object Oriented Programming Using Java

Course Objectives:

Practical implementations based on the OOPs features using JAVA.

Learning Outcomes:

Ability to:

1. Develop simple programs in Java using basic programming constructs.
2. Implement code using Java APIs related to standard packages.
3. Design the GUI applications.
4. Implement efficient data dictionaries using collection API.

Exercise-1

1) Write a Java Program that uses both recursive and non-recursive functions to print the n^{th} value of the Fibonacci sequence.
2) Write a Java Program that prints out all the prime numbers within a range.

Exercise -2

3) Write a Java Program that checks whether a given string is a palindrome or not. Ex: MALAYALAM is a palindrome.
4) Write a Java Program for sorting a given list of names in ascending order
5) Write a Java Program using String Tokenizer class, which reads a line of integers and then displays each integer and the sum of all integers

Exercise -3

6) Write a Java Program to demonstrate sequence of Constructor calling in a Hierarchy. Implement parameterized constructors also for hierarchic calls.
7) Write a Java Program check the compatibility for multiplication, if compatible multiply two matrices and find its transpose.

Exercise -4

8) Write a Java Program that depicts file stream API to check whether a directory exists,

directory creation, number of files in a directory, file length and file content

Exercise -5

9) Write a Java Program that serialize and De- serialize an Employee object with a specific state for name, id and designation properties. Use transient also for one property.
10) Write a Java Program that implements a simple client/server application. The client sends data to a server. The server receives the data, uses it to produce a result and then sends the result back to the client. The client displays the result on the console. For ex: The data sent from the client is the radius of a circle and the result produced by the server is the area of the circle

Exercise -6

11) Write a Java Program that implements stack or queue with overflow and underflow conditions as Exceptions. Create your own exception classes for Overflow and Underflow

Exercise -7

12) Write a Java Program that converts infix notation to post fix and prefix notations.
13) Write a Java Program that evaluates a post fix and prefix notations

Exercise -8

14) Write a Java Program that creates 3 threads by extending Thread class. First thread displays "Good Morning" every 1 sec, the second thread displays "Hello" every 2 seconds and the third displays "Welcome" every 3 seconds. (Repeat the same by implementing Run able)
15) Write a Java Program that correctly implements Producer-Consumer problem using the concept of Inter Thread Communication

Exercise -9

16) Write a Java Program that allows user to paint text, lines, rectangles and ovals (3 d objects if any)
17) Write a Java Program that depicts mouse and key board action events

Exercise -10

18) Write a Java Program that works as a simple calculator. Use a grid layout to arrange buttons for the digits and for the + - x / % operations. Add a text field to display the result..

Exercise -11

19) Write a Java Program to Sort the DVD list based upon title, year of release, and Rating using comparator and Comparable Interface

Exercise -12

20) Write a Java program to store list of Employee objects (Employee object will have

empId, first Name, last Name, address, salary, deptId etc…) and find out all employees for a given deptId and print the results to any output stream. [use Lists/ HashMaps/ Iterators]

21) Write a program to implement Tree Set (or its sub-interfaces NavigableSet and SortedSet).with the elements which are sorted and navigable. Using NavigableSet get the element in set which is <= to the given element. i.e. like when searching a telephone dictionary, you want to see all names starts with A or AR or ARU or ARUN.

In addition to the Listed Experiments above ask the students to implement the following Case Study.

We are not asked to solve a problem in this Case Study; rather, we are asked to produce a classes for the library. We have to create a test driver to be sure the class works properly, but we do not need to deliver it to the client. Thus our usual pattern of object-oriented problem solving should be appropriate here.

Case Study:

Retail Store Management System

The Home Appliances shop wants to automate the following processes:

1. Booking a product by customer,
2. Bill generation for customer
3. Inventory status check and updation

Home Appliances shop owns various models of products, for eg . LG Television T101, LG Refrigerator R601, LG Micro Oven M701 etc. The shop keeps stock of bulk of products in same model to meet the customer demand.

For eg:

20 pieces of Sony 24" TV is available in the shop store. The product Ids of each piece will be different but product name is same for all pieces. The customer should check the availability of a product before booking the product. As per the availability of the product in the retail store, the customer can book one or more products based on the demand of the customer. Booking has to be done based on Product names. The customer can book only one product type at a time.

Design and implement **Customer class, Product class and RetailStore class**.

Customer

Customer should know about his customerID, customerName, contactNumber and the details of products booked by the customer.

Following are the responsibilities of Customer class:

Customer (int customerID,String name,int contactNo):

The Constructor creates Customer object with the given id, name & contact no.

addBookedProduct

Public void addBookedProduct(Product obj) :

The method adds the product which is booked by the customer to his bookedProductList.

Product

Product should know about productID,productName,productStatus and productPrice.

(note: productStatus r epresents the status of the product. It can be "Available" or "Booked".)

Following are the responsibilities of Product class:

Product (int productID,String productName, double ProductPrice)

The constructor initializes the value of productID, name & price with the given values .Initially the status of the product is set to “Available”.

RetailStore

RetailStore class should know about the storeName, customers registered with the store and product details available in the store.

RetailStore holds following responsibilities. Implement the below mentioned responsibilities for RetailStore class.

RetailStore(String storeName): The constructor initialize the storeName with the given input value.

generateCustomerID

public int **generateCustomerID**()

The method generates the customerID by incrementing the value of previous customer’s ID by 1. The value of Customer Id is initially set to zero. For the first customer added to the system the ID should be 1,for the second customer, ID should be 2 and so on.

The method should return the ID generated for the customer.

generateProductID

public int **generateProductID**()

The method generates the productID by incrementing the value of previous product’s

ID by 1. The value of productId is initially set to zero. For the first product added to the system the ID should be 1,for the second product , ID should be 2 and so on.

The method should return the ID generated for the product.

addCustomer

public void **addCustomer**(String name,int contactNo)

The method creates a customer ID , with that it creates a customer object and then it

adds to the customerList of RetailStore.
Hint:To create the customerId,use generateCustomerID method.

addProduct

public void **addProduct**(String name,String status,double price)

The method creates a product ID , with that it creates a product object and then it adds to the productList of RetailStore Hint:To create the productId,use generateProductID method.

checkProductAvailability

public int **checkProductAvailability**(String ProductName)

The method returns the count of products which are in status “Available” with the given name .

bookProduct

public int **bookProduct**(int CustomerID,String ProductName, int NumberOfProduct)
The method checks for the availability of the given product in the productList of retail store.If the required number of products are available in the store , then that many products are booked which means that ,that many products are being added to the bookedProductList of customer.

If the required number of products are not available then the available quantity of products is being booked.
The method should return the total number of products booked.

If the customer tries to book 7 products and only 3 are available, it should book 3 and the method should return 3.

If the given product is not available in the product list or no stock available, it should return 0. If the given customer is not available in the customer list, then method should return-1.

calculateNetAmount

public double calculateNetAmount(int customerID,double discount) The method takes a customer ID and discount percentage as input parameters and calculates and returns the net amount that the customer need to pay for the entire booking he has made. Net amount= sum of price of all products - discount applicable (note: discout is applied to the sum of price of all products not on individual product price)

If the given customer is not available in the customer list, then the method should return -1.
Provide necessary getters and setters for all the classes.