DHANALAKSHMI SRINIVASAN ENGINEERING COLLEGE
(AUTONOMOUS)
(Approved by AICTE & Affiliated to Anna University, Chennai)
Re-Accredited by NAAC with 'A' Grade
Accredited by NBA for AERO, BME, CSE, ECE, EEE, IT & MECH.
PERAMBALUR-621212, TAMILNADU, INDIA.
Website: www.dsengg.ac.in

LABORATORY COURSE PLAN (2024 – 2025 ODD SEMESTERS)

LAB COURSE TITLE	OBJECT ORIENTED PROGRAMMING LABORATORY			
LAB COURSE CODE	U23ITP31			
LAB COURSE STRUCTURE	**LECTURE**	**TUTORIAL**	**PRACTICAL**	**CREDIT**
	0	0	4	2
REGULATION	**BRANCH**	**YEAR / SECTION**	**SEMESTER**	**ACADEMIC YEAR**
2023	IT	II / B & C	III	2024 - 2025
COURSE INCHARGE				

SYLLABUS

COURSE OBJECTIVE:

- The main learning objective of this course is to prepare the students for:
- To build software development skills using java programming for real-world applications.
- To understand and apply the concepts of classes, packages, interfaces, inheritance, exception handling and file processing.
- To develop applications using generic programming and event handling.

LIST OF EXPERIMENTS

1. Solve problems by using sequential search, binary search, and quadratic sorting algorithms (selection, insertion)
2. Develop stack and queue data structures using classes and objects.
3. Develop a java application with an Employee class with Emp_name, Emp_id, Address, Mail_id, Mobile no as members. Inherit the classes, Programmer, Assistant Professor, Associate Professor and Professor from employee class. Add Basic Pay (BP) as the member of all the inherited classes with 97% of BP as DA, 10 % of BP as HRA, 12% of BP as PF, 0.1% of BP for staff club funds. Generate pay slips for the employees with their gross and net salary.
4. Write a Java Program to create an abstract class named Shape that contains two integers and an empty method named print Area(). Provide three classes named Rectangle, Triangle and Circle such that each one of the classes extends the class Shape. Each one of the classes contains only the method print Area( ) that prints the area of the given shape.
5. Solve the above problem using an interface.
6. Implement exception handling and creation of user defined exceptions

7. Write a java program that implements a multi-threaded application that has three threads. First thread generates a random integer every 1 second and if the value is even, the second thread computes the square of the number and prints. If the value is odd, the third thread will print the value of the cube of the number.
8. Write a program to perform file operations.
9. Develop applications to demonstrate the features of generics classes.
10. Develop applications using JavaFX controls, layouts and menus.
11. Develop a mini project for any application using Java concepts.

TOTAL: 60 PERIODS

TEXT/REFERENCE BOOKS:

1.Herbert Schildt, "Java: The Complete Reference", 11th Edition, McGraw Hill Education, New Delhi,2019

2. Herbert Schildt, "Introducing JavaFX 8 Programming", 1st Edition, McGraw Hill Education, New Delhi, 2015

3.Cay S. Horstmann, "Core Java Fundamentals", Volume 1, 11 th Edition, Prentice Hall, 2018.

VIRTUAL LAB LINK:

W1: https://connect2compute.wordpress.com/java-lab-experiments/?utm_source=chatgpt.com
W2: https://www.studyglance.in/labprograms/r22javalabprograms.php?utm_source=chatgpt.com
W3: https://java-iitd.vlabs.ac.in/Objective.html?utm_source=chatgpt.com
W4: https://netajigandi.blogspot.com/2024/05/java-programming-lab.html?utm_source=chatgpt.com
W5: https://www.studocu.com/in/document/dr-apj-abdul-kalam-technical-university/object-oriented-programming-in-java-lab/2023-24-oops-with-java-lab-practical-list/91200534?utm_source=chatgpt.com
W6: https://www.studocu.com/in/document/amity-university/btech/java-lab-experiments/30794687?utm_source=chatgpt.com

EXP. NO.	NAME OF THE EXPERIMENTS	NO. OF PERIODS	CUMULATIVE PERIODS
1.	Solve problems by using sequential search, binary search, and quadratic sorting algorithms (selection, insertion)	8	8
2.	Develop stack and queue data structures using classes and objects	4	12
3.	Develop a java application with an Employee class with Emp_name, Emp_id, Address, Mail_id, Mobile no as members. Inherit the classes, Programmer, Assistant Professor, Associate Professor and Professor from employee class. Add Basic Pay (BP) as the member of all the inherited classes with 97% of BP as DA, 10 % of BP as HRA, 12% of BP as PF, 0.1% of BP for staff club funds. Generate pay slips for the employees with their gross and net salary	4	16

4.	Write a Java Program to create an abstract class named Shape that contains two integers and an empty method named print Area(). Provide three classes named Rectangle, Triangle and Circle such that each one of the classes extends the class Shape. Each one of the classes contains only the method print Area( ) that prints the area of the given shape	4	20
5.	Solve the above problem using an interface.	4	24
6.	Implement exception handling and creation of user defined exceptions	4	28
7.	Write a java program that implements a multi-threaded application that has three threads. First thread generates a random integer every 1 second and if the value is even, the second thread computes the square of the number and prints. If the value is odd, the third thread will print the value of the cube of the number.	4	32
8.	Write a program to perform file operations.	4	36
9.	Develop applications to demonstrate the features of generics classes.	8	44
10.	Develop applications using JavaFX controls, layouts and menus.	8	52
11.	Develop a mini project for any application using Java concepts	8	60

COURSE OUTCOME

At the end of the course the students would be able to

CO1: Design and develop java programs using object-oriented programming concepts
CO2: Design Interfaces and develop application using Interfaces
CO3: Develop simple applications using object-oriented concepts such as package, exceptions
CO4: Implement multithreading, and generics concepts
CO5: Create GUIs and event driven programming applications for real world problems
CO6: Develop applications using generic programming and event handlings.

CO-PO Mapping:

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO 10	PO11	PO12	PSO1	PSO2
CO1	3	1	2	0	2	0	0	0	0	0	0	0	2	2
CO2	1	1	2	0	2	0	0	0	0	0	0	0	2	2
CO3	2	2	1	0	2	0	0	0	0	0	0	0	2	2
CO3	1	2	1	0	1	0	0	0	0	0	0	0	2	2
CO4	2	2	1	0	1	0	0	0	0	0	0	0	2	2
CO5	2	1	2	0	2	0	0	0	0	0	0	0	2	2
CO6	2	1	1	0	2	0	0	0	0	0	0	0	2	2
AVG	2.16	1.57	1.43	0.00	1.71	0.00	0.00	0.00	0.00	0.00	0.00	0.00	2	2

MODEL LAB DETAILS

BATCH	REGISTER NO.	MODE OF LAB CONDUCT	DATE	TIMING
I		OFFLINE		
II		OFFLINE		
III		OFFLINE		
IV		OFFLINE		

LIST OF QUESTIONS

1. Define a class Student with attributes roll Number, name, and marks. Include methods to input student details and display them.
2. Create a class Rectangle with attributes length and width. Add methods to calculate and display the area and perimeter of the rectangle.
3. Write a program to create a Vehicle class with brand and model. Extend it with a Car class that adds an attribute number of Doors. Display all details.
4. Create a base class Animal with a method make Sound (). Extend it with Dog and Cat classes, overriding the method. Demonstrate polymorphism.
5. Write an abstract class Appliance with an abstract method turn On(). Implement it in subclasses Fan and Light.
6. Create an abstract class Account with an abstract method calculate Interest (). Implement it in Savings Account and Current Account.
7. Write a program that takes an integer array as input and divides all elements by a user-provided divisor. Handle exceptions for divide-by-zero and invalid input formats using try-catch blocks.
8. Create a program to validate the age of a user for eligibility to vote. If the age is less than 18, throw a custom exception Invalid Age Exception with an appropriate error message.
9. Write a Java program to create two threads: one to print even numbers and the other to print odd numbers within a given range.
10. Develop a program to implement a Producer-Consumer problem using threads and synchronization.
11. Write a program to read a text file line by line and count the number of words in each line. Display the output in the console.

12. Write a program to copy the contents of one file into another. Handle exceptions for file not found or input/output errors.
13. Create a generic class Box with methods to add an item and retrieve it. Demonstrate the functionality of the Box class for different data types like Integer, String, and Double.
14. Write a program to implement a generic method find Maximum that finds the largest element in an array of any data type.
15. Develop a simple Java FX application with a GUI to accept a user's name and display a personalized greeting when a button is clicked.
16. Create a Java FX application to simulate a basic calculator with buttons for digits and operations, displaying results in a label.

VI VA QUESTIONS

1. What is the difference between sequential search and binary search?
2. Why is binary search faster than sequential search? What is the precondition for binary search?
3. Explain the working of selection sort and insertion sort.
4. What is the time complexity of each sorting algorithm?
5. Which sorting algorithm is stable and why?
6. How do you handle duplicates in searching and sorting?
7. What is a stack? Explain LIFO principle.
8. What are the main operations in a stack? Describe them.
9. How is a queue different from a stack? Explain FIFO.
10. What are the types of queues?
11. How would you implement a queue using an array or linked list?
12. What are applications of stacks and queues?
13. What is inheritance in Java? Why is it useful?
14. What are the access specifiers used in your Employee class and why?
15. How did you calculate DA, HRA, PF, and staff club funds in your program?
16. What is method overriding and did you use it anywhere?
17. How do you generate the pay slip for employees in your program?
18. Explain how you used constructors in the inheritance hierarchy.
19. What is an abstract class in Java?
20. Why can't you instantiate an abstract class?
21. What is the purpose of the abstract method `printArea()` in your program?
22. How do subclasses implement the abstract method?
23. Can abstract classes have constructors? Explain.
24. What is the difference between abstract class an
25. What is an interface in Java?
26. How is an interface different from an abstract class?
27. Can you have method implementations inside an interface?
28. Why would you choose to use interfaces instead of abstract classes?
29. How do classes implement an interface?
30. What is exception handling in Java?
31. Explain the try-catch-finally block.

32. What are checked and unchecked exceptions?
33. How do you create a user-defined exception?
34. Why should you use exception handling in your programs?
35. What happens if an exception is not caught?
36. What is a thread in Java?
37. How do you create and start a thread?
38. What is the difference between extending Thread and implementing Runnable?

Google Class Code Details: B & C Section: oqdhilpl

Class Name: U23ITP31 - OBJECT ORIENTED PROGRAMMING LABORATORY

PREPARED BY **VERIFIED BY**

APPROVED BY

PRINCIPAL