22UCSL403	Object Oriented Programming Laboratory	(0-0-2) 1

Contact Hours: 26

Course Learning Objectives (CLOs): This laboratory course focuses on the following learning perspectives:

- Object Oriented (OO) concepts/philosophy and its benefits and drawbacks in system development.
- Basic features of Java programming language to implement Object Oriented (OO) Key concepts like ADT/Encapsulation, reusability (Inheritance/Composite Objects), polymorphism etc., and other core basic features.

Course Outcomes (COs):

Description of the Course Outcome: At the end of the course the student will be able to:		**Mapping to POs(1-12) / PSOs (13-16)**		
		Substantial Level (3)	**Moderate Level (2)**	**Slight Level (1)**
CO-1	**Prepare** an abstract data type for the given business scenario and **write** simple programs to represent ADT and **use** in the given application scenario.	13	1	-
CO-2	**Write** programs to solve given problem using different reusability features like inheritance and composite objects.	2, 14	1, 16	3
CO-3	**Write** a program to solve given problem using utility classes.	2, 14	1	3
CO-4	**Write** a program to solve given problem using abstract classes and differentiate with interfaces.	2, 14, 16	1	3
CO-5	**Write** a program to solve given problem using packages.	2, 14, 16	1	3
CO-6	**Write** a program to solve given problem using exception handling in construction of robust systems.	2, 14	1	3
CO-7	**Use** multithreading concept to solve conflicts due to interleaved execution of threads and write simple programs.	2, 14	1	3
CO-8	**Use** streams concept in developing system that needs facility for storage and retrieval of data.	2, 14	1	3
CO-9	**Design and Develop** GUI based system using applet, frames, events and other support available in AWT / Swings components.	2, 8, 14	1	3

POs/PSOs	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Mapping Level	2.0	3.0	1.0	-	-	-	-	3.0	-	-	-	-	3.0	3.0	-	2.3

Pre-requisites: Knowledge of: Registration/Completion of the course Object Oriented Programming.

Suggested Platforms:
Notepad (Non IDE), IDE (JCreator, Net Beans, Eclipse etc) in Windows OS and Linux OS

All programs should:

1. Be written to realize the Object Oriented Philosophy and core Java features.
2. Be written with Java Naming & Coding conventions and well documented.
3. Handle exceptions.
4. Be tested for all possible scenarios.

Course Contents:

Minimum one exercise to cover each course outcome specified above. Minimum 8 experiments to be completed by each student independently covering all course outcomes defined for this course. Course teacher has to publish list of experiments along with individual outcome for every experiments, on the first day of the semester. Examiner may set any problem based on the published term work during tests.

Reference Books:

1) Herbert Schildt, "Java: The Complete Reference: 7th Edition, Tata McGraw Hill, 2007.
2) Kathy Sierra & Bert Bates, "Head First Java", 2nd Edition, O'Reilly, 2009
3) Patrick Niemeyer & Daniel Leuck, "Learning Java", 4th Edition, O'Reilly, 2013
4) Laura Lemay & Charles L. Perkins, "Teach Yourself Java in 21 Days", 7th Edition, Sams Publishing, 2016