Course Code	CSE 201
Course Name	Advanced Programming
Credits	4
Course Offered to	UG
Course Description	The Advanced Programming is a successor to the Introduction of Programming course. The main goal of this course is to prepare students to the challenge of building large-scale programs which multiple functional components, some of which could be designed/implemented independently. The course will use Java to to introduce students to concepts of object orientation, reusable code design, test-driven development, programming to an application-programming-interface, pattern oriented program design and implementation etc. At the end of the course the students are expected to be able to work in teams in order to develop large application programs starting from a reasonably well-defined application design with multiple independent components with well-defined interfaces. NOTE: The course is not meant to teach the features of Java programming language

Pre-requisites

Pre-requisite (Mandatory)	Pre-requisite (Desirable)	Pre-requisite(other)
CSE101, CSE102		

*Please insert more rows if required

Post Conditions*(For suggestions on verbs please refer the second sheet)

CO1	CO2	CO3	CO4	CO5
Student are able to demonstate knowledge of the basic principles of the object oriented development process and apply this understanding to the analysis and design of solutions for small to medium scale problems.	Implement basic event-driven programming, exception handling, and threading.	Students are able to analyze the problem in terms of use cases and create object oriented design for it. Students are able to present the design in UML or other related tools.	Students are able to select and use a few key design pattern to solve a given problem in hand.	Students are able to use common tools for testing (e.g., Junit), debugging, and source code control as an integral part of program development.

Weekly Lecture Plan

Week Number	Lecture Topic	COs Met	Assignment/Labs/Tutorial
1	• Introduction to Object Oriented Paradigm • Examples of OOP • Data encapsulation, modularity, code reuse	CO1	

2	• Inheritance and access modifiers • Object oriented modeling of real-world • Polymorphic behaviour of objects • Polymorphic data-structures	CO1
3	• Development of testcases, unit testing of code - JUnit • Test driven program development - with and without an IDE • Debugging ang profiling of programs • Introduction to version control systems, tracking of bugs, program maintenance • Importance of coding style	CO5
4	• Introduction to UML	C03

5	• Introduction to Java Standard Class Library, package organization • Abstract data-types (ADT) • Interfaces, abstract methods and classes	CO1
6	• Introduction to GUI programming • Event driven programming	CO2
7	• Introduction to defensive and secure programming • Exception handling • Assertions	CO2
8	• Achieving efficiency in programs - memory utilization, processor utilization through multithreading • Thread pool	CO2
9	• Mutual exclusion • Locks	CO2
10	• Introduction to pattern-oriented program design	CO4
11	• Some design patterns and examples based on pattern-oriented program design	C04
12	• More Design Patterns • Model-View-Controller pattern and its use in application Development	C04
13	Spill Over	

Weekly Lab Plan

Week Number	Laboratory Exercise	COs Met	Platform (Hardware/Software)
	Aligned with the lectures		

*Please insert more rows if required

Assessment Plan

Type of Evaluation	% Contribution in Grade
Quiz	10
Laboratory	20
Project	20
Mid-sem	20

End-sem	30

*Please insert more row for other type of Evaluation

Resource Material	
Type	**Title**
Textbook	Core Java - Volumes I and II
Reference	Design Patterns: Elements of Reusable Object-Oriented Software
Reference	Program Development in Java - Abstraction, Specification, and Object-Oriented Design.