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**
Students are able to demonstrate the knowledge of basic principles of Object Oriented Programming such as encapsulation (classes and objects), interfaces, polymorphism and inheritance; by implementing programs ranging over few hundreds lines of code.	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.	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 • Encapsulation • Identifying classes and objects • Working with objects	CO1	
2	• Class relationships * Interfaces	CO1, C03	
3	• Polymorphism • Inheritance	CO1, C03	
4	• Abstract class and methods • Object class • Comprable and Comparator interfaces	C01	

5	* Generic programming * Collection framework	CO1	
6	• Introduction to defensive programming • Exception handling • Assertions * Debugging programs	CO2, C05	
7	* Unit testing * Junit * Source code control	CO5	
8	* UML diagrams * Introduction to even driven programming using JavaFX	CO2, C03	
9	* Introduction to processes and threads * Multithreaded programming	CO2	
10	* Thread pool * Mutual exclusion • Locks	CO2	
11	* Design patterns	C04	
12	* Some more design patterns	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.