Course Code	CSE600A/ECE600A
Course Name	Object Oriented Programming and Design
Credits	2
Course Offered to	PG
Course Description	It is a 2 credit postgraduate level core course in advanced object oriented programming, designed to improve programming skills, and ability to use contemporary software development practices and tools. The course curriculum focuses on large software development, collaborative development environment, automatic testing and deployment, common libraries and continuous integration. It also aims to enhance the quality of design in terms of scalability and maintainability with design principles and design patterns. Along with course, students get an opportunity to learn development tools like IDEs and common frameworks.

Pre-requisites		
Pre-requisite (Mandatory)	Pre-requisite (Desirable)	Pre-requisite(other)
	CSE102 Data Structures & Algorithms	
	CSE101 Intro to Programming	

*Please insert more rows if required

Post Conditions*(For suggestions on verbs please refer the second sheet)			
CO1	**CO2**	**CO3**	**CO4**
Students are able to develop object oriented software using common design patterns (of size around 500 - 1000 lines).	Students are able to effectively use various software eng tools like application development environments, version control and collaboration, continuous integration, automation testing and deployment.		

Weekly Lecture Plan			
Week Number	**Lecture Topic**	**COs Met**	**Assignment/Labs/Tutorial**
Week 1	Introduction to course content, assesment structure, grading policy), Review of OOP basics	CO1	There will be a single big case study divided into 6 modules or assignments due every two weeks to implement and analyse the OOPD concepts covered in the lectures. Tutorial and lab will be based on the implementations of the concepts covered in the lectures in either Java (for CSE) or Python (ECE/CB)
Week 2 - Week 4	Advanced OOP concepts: Inheritance and Polymorphism, Abstract classes and interfaces, Name spaces, Methods and operator overloading, Multiple Inheritance, Exception handling; Version control, Connecting to databases	CO1	
Week 5	Design Principles - Introduction to UML, design patterns (Model-View-Controller)	CO1	
Week 6	Automation testing - unit and integration testing	CO2	
Week 7	Code review	CO1 and CO2	
Week 8 and Week 9	User interfaces and GUI	CO2	
Week 10	Performance and scalability	CO2	
Week 11 and Week 12	Collaborative programming and Continuous integration	CO2	
Week 13	Revision of topics, Q& A	CO1 & CO2	

*Please insert more rows if required

Weekly Lab Plan			
Week Number	**Laboratory Exercise**	**COs Met**	**Platform (Hardware/Software)**
Week 1	Introducring IDEs for Java (Eclipse) and Python (Pycharm), Lab exercises based on topics covered in Week 1	CO1 & CO2	Programming in Java or Python with Eclipse/Pycharm
Week 2 - Week 4	Lab exercises based on topics covered on week 2 to week 4. Introducing GitHub for version control	CO1 & CO2	Programming in Java or Python with Eclipse/Pycharm

Week 5	Lab exercises based on topics covered in week 5	CO1 & CO2	Programming in Java or Python
Week 6	Lab exercises based on topics covererd in week 6, introducing TestNG/Mock/Junit (for CSE) and PyUnit (for ECE/CB)	CO1 & CO2	Programming in Java or Python
Week 7	Lab/tut covering topics covered in week 7, can work on their assignments (projects)	CO1 & CO2	Programming in Java or Python
Week 8 & 9	Lab exercises covering topics in week 8 and 9.	CO1 & CO2	Programming in Java or Python
Week 10	Lab exercises covering topics in week 10.	CO1	Programming in Java or Python
Week 11 & 12	Lab exercises covering topics in week 11 & 12.	CO1 & CO2	Programming in Java or Python
Week 13	Demo of complete application developed through the assignments	CO1 & CO2	Programming in Java or Python

*Please insert more rows if required

Assessment Plan	
Type of Evaluation	% Contribution in Grade
Assignment	60
Quiz	15
End-sem	25
*Please insert more row for other type of Evaluation	

Resource Material	
Type	**Title**
Textbook	1. Design Patterns: Elements of Reusable Object-Oriented Software by Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides
Textbook	1. Object Oriented Programming – Learning Python by Mark Lutz, David Ascher
Textbook	1. Core Java text books such as Absolute Java by Walter Savitch
Resource Material	https://www.ntu.edu.sg/home/ehchua/programming/index.html
Resource Material	https://python-textbok.readthedocs.io/en/1.0/Python_Basics.html
Resource Material	Maven Documentation (https://maven.apache.org/guides/index.html)
Resource Material	TestNG documentation (http://testng.org/doc/documentation-main.html)
Resource Material	JMock Documentation (http://www.jmock.org/cookbook.html)
Resource Material	Continuous integration for Python (https://www.fullstackpython.com/continuous-integration.html)