Object Oriented Programming & UML
Code: IT401
Contacts: 3L+1T
Credits: 4

Prerequisites of Object Oriented Programming & UML:
The fundamental point in learning programming is to develop the critical skills of formulating programmatic solutions for real problems. It will be based on basic knowledge of algorithms and procedural programming language. Once the basic skill of writing programs using loop, methods and arrays will be clear then the student can develop object oriented software using class encapsulation and inheritance.

Module-1: [10L]
Introduction:
Why object orientation, History and development of object oriented programming language, concepts of object oriented programming language. [1L] Difference between OOP and other conventional programming – advantages and disadvantages. [1L] Data types, variables. Array, operators. [1L] String, I/O. [1L] Control statements. [1L]

Object oriented design:
Major and minor elements, class fundamentals. [1L]; Declaring objects, instantiation of class, introducing methods. [1L]; Constructing objects using constructor. [1L]; Static variable, constants. [1L]; Visibility modifiers. [1L]

Module-2: [8L]
Object Properties: Introduction to basic features of a class (encapsulation, polymorphism etc) [1L]; Data field encapsulation. [1L]; Passing objects to methods. [1L]; Array of objects, 'This' keyword [1L]; Relationships among objects: aggregation, composition, dependency, links. [1L]; Relationship among classes: association, aggregation. [1L]
Meta class, meta object. [1L]; Grouping constructs. [1L]

Module-3: [11L]
Basic concepts of object oriented programming using Java:
Using objects as parameters, closure look at argument passing, returning objects. [1L]; Introducing access control, Final keyword, garbage collection, Nested and inner classes. [1L]; Class abstraction and encapsulation, Overloading of methods (overloading of constructor). [1L]; Super class, subclasses, super keyword, inheritance, types, member access.[1L]; Multilevel hierarchy, process of constructor calling in inheritance. [1L]; Overriding methods, overriding vs. overloading, polymorphism. [1L]; Abstract class, interface & comparison between abstract class and interface [1L]; Packages, importing packages. [1L]; Exception handling basics, types, using try &catch, throw, throws & finally. [1L]; Threading, synchronization & priorities, thread class, creating thread. [1L]; Basic applet programming. Life cycle. [1L];

Module-4: [8L]
Fundamentals of Object Oriented design in UML:
Introduction to UML: Why Modeling, Overview of UML, Conceptual Model, Architecture of UML [1L]; UML Modeling Types: Structural Modeling, Behavioral Modeling, Architectural Modeling [1L]; Basic Notations in UML [1L]; Class Diagram [1L]; Interaction and Collaboration Diagrams. [1L]; Sequence Diagram. [1L]; State chart Diagram and Activity Diagram. [1L]; Implementation Diagram and UML extensibility- model constraints.[1L]

Textbooks/References:
1. Rambaugh, James Michael, Blaha-"Object Oriented Modelling and Design"-Prentice Hall, India
2. Ali Bahrami,-"Object Oriented System Development"-Mc Graw Hill
3. Patrick Naughton, Herbert Schildt-"The complete reference-Java2"-TMH
4. Sourav Sahay-"Object-Oriented Programming with C++"-Oxford

5. Jason T. Roff, UML: A Beginner's Guide, TMH
6. Grady Booch, Ivar Jacobson, James Rumbaugh, , “The Unified Modeling Language Reference Manual”, Pearson Ed.
7. Blaha, Rumbaugh, "Object-Oriented Modeling and Design with UML", Pearson Ed.