MCA- C204 Object Oriented Technology

L	T	P	C
4	0	0	4

Course objective:

The course will introduce standard tools and techniques for software development, using object-oriented approach, use of a version control system, an automated build process, an appropriate framework for automated unit and integration tests.

Course outcomes:

1. Specify simple abstract data types and design implementations, using abstraction functions to document them.
2. Recognize features of object-oriented design such as encapsulation, polymorphism, inheritance, and composition of systems based on object identity.
3. Name and apply some common object-oriented design patterns and give examples of their use.

Introduction: Fundamental concepts, Objects and legacy systems, Procedural vs OO programming, Object data, object behavior, creating objects, attributes, methods, messages, encapsulation and data hiding, super classes and sub classes, abstraction, Is-a relationship, polymorphism, abstraction, Has-a relationship

Objects: The interface, the implementation, determining the users, object behavior, environmental constraints, identifying the public interfaces, identifying the implementation

Advanced concepts: Constructors, error handling, importance of scope, operator overloading, multiple inheritance, object operations

Anatomy of a class: Name, comments, attributes, constructors, assurors, public interface methods, private implementation methods

Class design guidelines: modeling real world systems, identifying the public interfaces, designing robust constructors, designing error handling into a class, designing reuse in mind, designing extensibility in mind, designing maintainability in mind, using object persistence

Designing with objects: Performing the proper analysis, developing a statement of work, gathering the requirements, developing the prototype, identifying the classes, determining the responsibility of a class, creating a class model, prototyping the user interface, object wrappers

Inheritance and composition: Reusing objects, generalization and specialization, design decisions, representing composition with UML, object responsibility, abstract classes, virtual methods and protocols

Frameworks and reuse: Framework, contract, abstract classes, interfaces, making a contract, an E-business example

Object-oriented design: Composition relationships, building in phases, types of compositions, avoiding dependencies, cardinality

Creating object models: What is UML? Structure of a class diagram, attributes and methods, access designations, inheritance, interfaces, composition, cardinality

Design patterns: Why design patterns, model-view-controller, types of design patterns, anti-patterns

[illegible]

HEAD
Department of Computer Science
Gurukul Kangri Vishwavidyalaya
Haridwar (UK) - 249404

Recommended Books:
1. Weisfeld M., The Object-Oriented Thought Process, Addison-Wesley Professional
2. Shalloway A., Trott J., Design Patterns Explained: A New Perspective on Object-oriented Design, Addison-Wesley
3. Fowler M., UML Distilled, Addison-Wesley

[illegible]

HEAD
Department of Computer Science
Gurukul Kangri Vishwavidyalaya
Haridwar (UK) - 249404