



# INSTITUTE OF AERONAUTICAL ENGINEERING

(Autonomous)

Dundigal - 500 043, Hyderabad, Telangana

## COURSE CONTENT

OBJECT ORIENTED PROGRAMMING								
<b>I Semester: AE / ME / CE / ECE / EEE / CSE / CSE (AI &amp; ML) / CSE (DS) / CSE (CS) / IT</b>								
Course Code	Category	Hours / Week			Credits	Maximum Marks		
		L	T	P		C	CIA	SEE
ACSD01	Foundation	3	0	0	3	40	60	100
<b>Contact Classes: 48</b>		<b>Tutorial Classes: Nil</b>		<b>Practical Classes: Nil</b>		<b>Total Classes: 48</b>		
<b>Prerequisites: There are no prerequisites to take this course.</b>								

### I. COURSE OVERVIEW:

The course provides a solid foundation in object-oriented programming concepts in using them. It includes concepts object-oriented concepts such as information hiding, encapsulation, and polymorphism. It contrasts the use of inheritance and composition as techniques for software reuse. It provides an understanding of object-oriented design using graphical design notations such as Unified Modelling Language (UML) as well as object design patterns.

### II. COURSES OBJECTIVES:

#### The students will try to learn

- I. The fundamental concepts and principles of object-oriented programming in high-level programming languages.
- II. The advanced concepts for developing well-structured and efficient programs that involve complex data structures, numerical computations, or domain-specific operations.
- III. The design and implementation of features such as inheritance, polymorphism, and encapsulation for tackling complex problems and creating well-organized, modular, and maintainable code.
- IV. The usage of input/output interfaces to transmit and receive data to solve real-time computing problems.

### III. COURSE OUTCOMES:

#### At the end of the course, students should be able to:

- CO 1 Interpret the features of object-oriented programming languages, comparison, and evaluation of programming languages.
- CO 2 Model the real-world scenario using class diagrams and exhibit communication between objects.
- CO 3 Estimate the need for special functions for data initialization.
- CO 4 Outline the features of object-oriented programming for binding the attributes and behavior of a real-world entity.
- CO 5 Use the concepts of streams and files that enable data management to enhance programming skills.
- CO 6 Develop contemporary solutions to software design problems using object-oriented principles.

### IV. COURSE CONTENT:

#### MODULE - I: Object-oriented concepts (09)

Objects and legacy systems, procedural versus Object-oriented programming, top-down and bottom-up approaches and their differences, benefits of OOP, applications of OOP, features of OOP.

**Abstraction:** Layers of abstraction, forms of abstraction, abstraction mechanisms.

#### MODULE - II: Classes and objects (09)

**Classes and objects:** Object data, object behaviors, creating objects, attributes, methods, messages, creating class diagrams.

**Access specifiers and initialization of class members:** Accessing members and methods, access specifiers - public, private, protected, memory allocation. Static members, static methods.

### **MODULE - III: Special member functions and overloading (09)**

**Constructors and destructors:** Need for constructors and destructors, copy constructors, dynamic constructors, parameterized constructors, destructors, constructors and destructors with static members.

**Overloading:** Function overloading, constructor overloading, operator overloading - rules for overloading operators, overloading unary and binary operators, friend functions.

### **MODULE – IV: Inheritance and polymorphism (09)**

**Inheritance:** types of inheritance, base class, derived class, usage of final, ambiguity in multiple and multipath inheritance, virtual base class, overriding member functions, order of execution of constructors and destructors.

**Polymorphism and virtual functions:** Virtual functions, pure virtual functions, abstract classes, introduction to polymorphism, static polymorphism, dynamic polymorphism.

### **MODULE –V: Console I/O and working with files (09)**

**Console I/O:** Concept of streams, hierarchy of console stream classes, unformatted I/O operations, managing output with manipulators.

**Working with files:** Opening, reading, writing, appending, processing, and closing different types of files, command line arguments.

### **V. TEXTBOOKS:**

1. Matt Weisfeld, *The Object-Oriented Thought Process*, Addison Wesley Object Technology Series, 4<sup>th</sup> Edition, 2013.

### **VI. REFERENCE BOOKS:**

1. Timothy Budd, *Introduction to object-oriented programming*, Addison Wesley Object Technology Series, 3<sup>rd</sup> Edition, 2002.
2. Gaston C. Hillar, *Learning Object-Oriented Programming*, Packt Publishing, 2015.
3. Kingsley Sage, *Concise Guide to Object-Oriented Programming*, Springer International Publishing, 1<sup>st</sup> Edition, 2019.
4. Rudolf Pecinovsky, *OOP - Learn Object Oriented Thinking and Programming*, Tomas Bruckner, 2013.
5. Grady Booch, *Object-oriented analysis and design with applications*, Addison Wesley Object Technology Series, 3<sup>rd</sup> Edition, 2007.

### **VII. ELECTRONICS RESOURCES:**

1. <https://docs.oracle.com/javase/tutorial/java/concepts/>
2. <https://www.w3schools.com/cpp/>
3. <https://www.edx.org/learn/object-oriented-programming/>
4. <https://www.geeksforgeeks.org/introduction-of-object-oriented-programming/>

### **VIII. MATERIALS ONLINE**

1. Course template
2. Tutorial question bank
3. Tech talk topics
4. Open-ended experiments
5. Definitions and terminology
6. Assignments
7. Model question paper – I
8. Model question paper – II
9. Lecture notes
10. PowerPoint presentation
11. E-Learning Readiness Videos (ELRV)