

**Name of Institute: Indus Institute of Technology & Engineering**

**Name of Faculty: Zalak Trivedi**

**Course code: CE0316**

**Course name: Object Oriented Programming with UML**

Pre-requisites: -

Knowledge of C language will be useful.

Credit points: 4

Offered Semester: III

**Course coordinator**

Full name: Zalak Trivedi

Department with sitting location: CE dept, 4<sup>th</sup> floor Bhanwar Building.

Telephone:

Email: zalaktrivedi.ce@indusuni.ac.in

Consultation times:

**Monday 11.55am to 2:00pm**

**Friday 01.30pm to 2.30pm**

**Course lecturer**

Full name: Zalak Trivedi

Department with sitting location: CE dept, 4<sup>th</sup> floor Bhanwar Building.

Telephone:

Email: zalaktrivedi.ce@indusuni.ac.in

Consultation times:

**Monday 11.55am to 2:00pm**

**Friday 01.30pm to 2.30pm**

Students will be contacted throughout the session via mail with important information relating to this course.

### Course Objectives

1. Perform object oriented programming to develop solutions to problems demonstrating usage of control structures, modularity, I/O. and other standard language constructs.
2. Demonstrate adeptness of object oriented programming in developing solutions to problems demonstrating usage of data abstraction, encapsulation, and inheritance.
3. Demonstrate ability to implement one or more patterns involving realization of an abstract interface and utilization of polymorphism in the solution of problems which can take advantage of dynamic dispatching.
  
4. Learn syntax, features of, and how to utilize the Standard Template Library. Learn other features of the C++ language including templates, exceptions, forms of casting, conversions, covering all features of the language. Learn features of the language which can be problematic with execution time or space and some techniques to resolve them. Learn features of the language which are non-deterministic, should not be utilized in hard real-time systems, and techniques for replacing those features. Learn the C++ language changes and Boost library.

### Course Outcomes (CO)

By participating and understanding all facets of this course a student will be able to:

1. To build an **understanding** of important concepts of object oriented programming like object and class, Encapsulation, inheritance and polymorphism.
2. To **understand** the knowledge of writing the skeleton of C++ program.
3. To **create** C++ programs using objects, classes, I/O, tokens.
4. **Understanding** the features of C++ like type conversion, I/O streams and files.

5. To **create** programs for real life problems and projects based on this.
6. To **apply** the advance features like templates, exception and UML diagrams.

## Course Outline

<b>UNIT-I</b>	<b>[8 hours]</b>
<b>INTRODUCTION TO C++</b>	
Concepts of OOP: Introduction OOP, Procedural Vs. Object Oriented Programming, Principles of OOP, Benefits and applications of OOP C++Basics: Overview, Program structure, namespace, identifiers, variables, constants, enum, operators, typecasting, control structures C++ Functions: Simple functions, Call and Return by reference, Inline functions, Macro Vs. Inline functions, Overloading of functions, default arguments, friend functions	

<b>UNIT-II</b>	<b>[8 hours]</b>
<b>Objects and classes</b>	
Basics of object and class in C++, Private, protected and public Members, static data and static function,	
<b>Constructors and their types</b> , Destructors, Arrays & Strings: A standard C++ string class.	
<b>Operator Overloading</b> : Overloading unary and binary operators, Operator Overloading with friend function, Data Conversion, type conversion, class to class, basic to class, class to basic	

### UNIT-III

[8 hours]

#### Concept of Inheritance

Types of inheritance: single, multiple, multilevel, hierarchical, hybrid, protected members, overriding, virtual base class, constructor in derived classes

**Polymorphism:** Pointers in C++, Pointers and Objects, this pointer, virtual and pure virtual functions, implementing polymorphism

**I/O management:** Concept of streams, cin and cout objects, C++ stream classes, Unformatted and formatted I/O, manipulators

### UNIT-IV

[8 hours]

#### File management:

File stream, C++ File stream classes, File management functions, File modes, Binary and random files

#### Object-oriented Design

Object modeling using UML, Three models, Class Model (Object and Class Diagram), State model (state Diagram) and Interaction model (Use case diagrams, Activity diagrams, Interaction diagrams).

#### Method of delivery

Chalk and Board, PowerPoint presentation

#### Study time

3 Hours theory, 2 Hours practical

#### CO-PO Mapping (PO: Program Outcomes)

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	2	-	-	-	-	-	-	-	-	-	-	-
CO2	3	1	1	-	-	-	-	-	-	-	-	-
CO3	3	3	2	-	-	-	-	-	-	-	-	-
CO4	2	1	-	-	-	-	-	-	-	-	-	-
CO5	3	3	2	-	-	-	-	-	-	-	-	-

CO6	3	3	2	-	-	-	-	-	-	-	-	-
-----	---	---	---	---	---	---	---	---	---	---	---	---

**Blooms Taxonomy and Knowledge retention (For reference)**  
 (Blooms taxonomy has been given for reference)

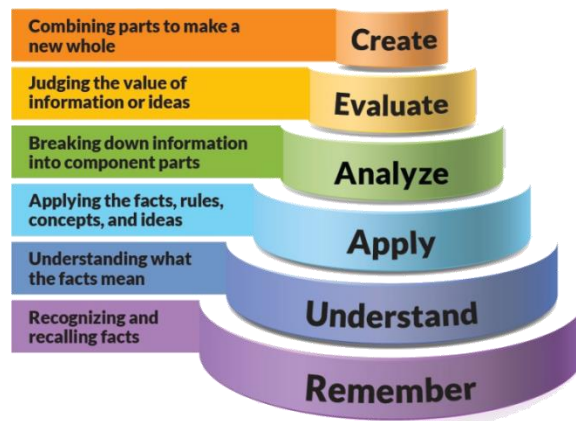


Figure 1: Blooms Taxonomy

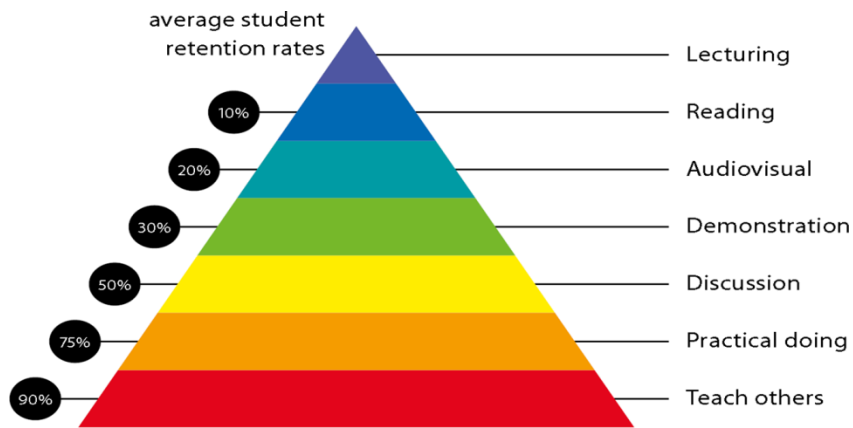


Figure 2: Knowledge retention

**Graduate Qualities and Capabilities covered**  
 (Qualities graduates harness crediting this Course)

General Graduate Qualities	Specific Department of _____ Graduate Capabilities
<p><b>Informed</b></p> <p>Have a sound knowledge of an area of study or profession and understand its current issues, locally and internationally. Know how to apply this knowledge. Understand how an area of study has developed and how it relates to other areas.</p>	<p><b>1 Professional knowledge, grounding &amp; awareness</b></p>
<p><b>Independent learners</b></p> <p>Engage with new ideas and ways of thinking and critically analyze issues. Seek to extend knowledge through ongoing research, enquiry and reflection. Find and evaluate information, using a variety of sources and technologies. Acknowledge the work and ideas of others.</p>	<p><b>2 Information literacy, gathering &amp; processing</b></p>
<p><b>Problem solvers</b></p> <p>Take on challenges and opportunities. Apply creative, logical and critical thinking skills to respond effectively. Make and implement decisions. Be flexible, thorough, innovative and aim for high standards.</p>	<p><b>4 Problem solving skills</b></p>
<p><b>Effective communicators</b></p> <p>Articulate ideas and convey them effectively using a range of media. Work collaboratively and engage with people in different settings. Recognize how culture can shape communication.</p>	<p><b>5 Written communication</b></p>
	<p><b>6 Oral communication</b></p>
	<p><b>7 Teamwork</b></p>
<p><b>Responsible</b></p> <p>Understand how decisions can affect others and make ethically informed choices. Appreciate and respect diversity. Act with integrity as part of local, national, global and professional communities.</p>	<p><b>10 Sustainability, societal &amp; environmental impact</b></p>

**Practical work:**

1	<ol style="list-style-type: none"> <li>1. Write a program to find factorial using functions.</li> <li>2. Write a program to get the student details like student name and roll no using class.</li> <li>3. Write a program to find area of a circle, square and rectangle.</li> </ol>	Basic knowledge of Programming
2	<ol style="list-style-type: none"> <li>1. Write a program to use default arguments.</li> <li>2. Write a program to use return by reference.</li> <li>3. Write a program for function overloading.</li> </ol>	Basic knowledge of UDF
3	<ol style="list-style-type: none"> <li>1. Write programs of sum of two nos. using inline function.</li> <li>2. Write a program to pass object as arguments in function.</li> <li>3. Write a program to use static member function.</li> </ol>	Basic knowledge of objects n classes
4	<ol style="list-style-type: none"> <li>1. Create a class student which stores the detail about roll no. , name, marks of five subjects. The class must have following:           <ul style="list-style-type: none"> <li>• Get() to accept value of the data members</li> <li>• Display() to display values of data members</li> </ul> </li> <li>2. Write a program to use friend function.</li> <li>3. Write a program which shows the use of scope resolution operator.</li> </ol>	Basic knowledge of friend function
5	<ol style="list-style-type: none"> <li>1. Write a program to illustrate the use of this pointer.</li> <li>2. Write a program to swap two values from two different classes using friend function.</li> </ol>	Basic knowledge of pointers
6	<ol style="list-style-type: none"> <li>1. Write a program to call member functions of class in the main function using pointer to object and pointer to member function.</li> <li>2. Using friend function, find maximum no. from given two nos. from two different classes. Write all necessary functions and constructors for the program.</li> <li>3. Using friend function, find the average nos. from three different classes. Write all necessary member functions and constructors</li> </ol>	Basic knowledge of functions

	for the classes.	
7	<ol style="list-style-type: none"> <li>1. Write a program using default constructor.</li> <li>2. Write a program using parameterized constructor.</li> </ol>	Basic principle of character constructor
8	<ol style="list-style-type: none"> <li>1. Write a program to demonstrate the use of destructor.</li> <li>2. Write a program for operator overloading using friend function.</li> <li>3. Write a program to demonstrate the use of type casting.</li> </ol>	Basic knowledge of type casting
9	<ol style="list-style-type: none"> <li>1. Write a program to demonstrate implementation of private/public mode inheritance.</li> <li>2. Write a program to demonstrate implementation of single inheritance.</li> <li>3. Write a program for multiple inheritance.</li> </ol>	Basic knowledge of inheritance
10	Demonstrate UML diagrams	Basic Knowledge of UML

### Lecture/tutorial times

(Give lecture times in the format below)

<b>Lecture</b>	<b>3CS(A)</b>	<b>11.10 to 12.10</b>	<b>Monday</b>
<b>Lecture</b>	<b>3CS(A)</b>	<b>10.00 to 11.00</b>	<b>Tuesday</b>
<b>Lecture</b>	<b>3CS(B)</b>	<b>02.00 to 03.00</b>	<b>Tuesday</b>
<b>Lecture</b>	<b>3CS(A)</b>	<b>10.00 to 11.00</b>	<b>Wednesday</b>

### Attendance Requirements

The University norms states that it is the responsibility of students to attend all lectures, tutorials, seminars and practical work as stipulated in the course outline. Minimum attendance requirement as per university norms is compulsory for being eligible for semester examinations.

### Text books



1. Object oriented Programming with C++ ,Balaguruswamy, Tata Mcgraw Hill Publication Co. Ltd 2000.
2. Object oriented programming in turbo C++ ,RobbetLofre, Galgotia Publication Pvt Ltd. 1994.

### Reference Books:

1. The Complete Reference C++ , Fourth Edition , Herbert Schildt , Tata Mcgraw Hill Publication.
2. The C++ programming language , BjarneStroustrup ,Addison

### Additional Materials

Web Resource

<https://nptel.ac.in/courses/106105082/>

<https://nptel.ac.in/downloads/106105080/>

### ASSESSMENT GUIDELINES

Your final course mark will be calculated from the following:

<b>CIE-Theory (60 Marks)</b>  <b>Attendance: 05 Marks</b>  <b>M.S.E: 40 Marks</b>  <b>Presentation: 05 Marks</b>  <b>Assignment: 10 Marks</b>	<b>CIE-Practical (60 Marks)</b>  <b>Class Regularity: 10 Marks</b>  <b>Practical Test: 20 Marks</b>  <b>Practical File: 10 Marks</b>  <b>Minor Project: 20 Marks</b>
<b>ESE-Theory-40 Marks</b>	<b>ESE-Practical-40 Marks</b>
<b>Total: 200 Marks</b>	

Students who receive an overall mark less than 40% in internal component or less than 40% in the end semester will be considered for supplementary assessment in the respective components (i.e internal component or end semester) of semester concerned. Students must make themselves available during the supplementary examination period to take up the respective components (internal component or end semester) and need to obtain the required minimum 40% marks to clear the concerned components.

### Practical Work Report/Laboratory Report:

A report on the practical work is due the subsequent week after completion of the class by each group.

### Late Work

Late assignments will not be accepted without supporting documentation. Late submission of the reports will result in a deduction of -% of the maximum mark per calendar day

### Format

All assignments must be presented in a neat, legible format with all information sources correctly referenced. **Assignment material handed in throughout the session that is not neat and legible will not be marked and will be returned to the student.**

### Retention of Written Work

Written assessment work will be retained by the Course coordinator/lecturer for two weeks after marking to be collected by the students.

### University and Faculty Policies

Students should make themselves aware of the University and/or Faculty Policies regarding plagiarism, special consideration, supplementary examinations and other educational issues and student matters.

**Plagiarism** - Plagiarism is not acceptable and may result in the imposition of severe penalties. Plagiarism is the use of another person's work, or idea, as if it is his or her own - if you have any doubts at all on what constitutes plagiarism, please consult your Course coordinator or lecturer. Plagiarism will be penalized severely.

*Do not copy the work of other students.*

*Do not share your work with other students (except where required for a group activity or assessment).*

### Course schedule (subject to change)

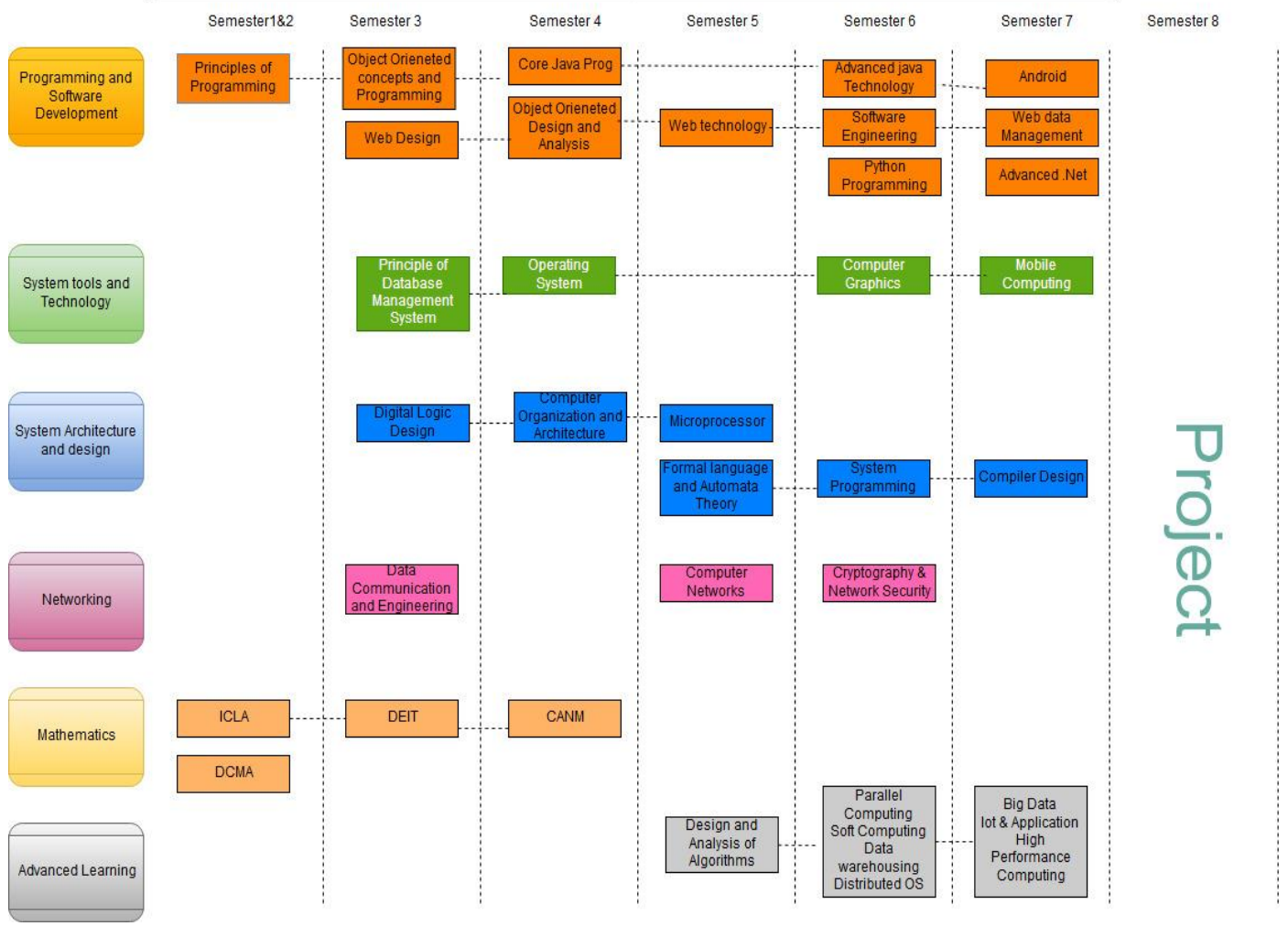
(Mention quiz, assignment submission, breaks etc as well in the table under the Teaching Learning Activity Column)

Week #	Topic & contents	CO Addressed	Teaching Learning Activity (TLA)
Week 1	Concepts of OOP: Introduction OOP, Procedural Vs. Object Oriented Programming, Principles of OOP, Benefits and applications of OOP	I	Chalk & Board, Discussion

Week 2	C++Basics: Overview, Program structure, namespace, identifiers, variables, constants, enum, operators, typecasting, control structures C++ Functions: Simple functions, Call and Return by reference, Inline functions, Macro Vs. Inline functions	<b>I</b>	Presentation, Chalk & Board
Week 3	Overloading of functions, default arguments, friend functions	<b>I</b>	Presentation, Chalk & Board
Week 4	<b>Objects and classes:</b> Basics of object and class in C++, Private, protected and public Members,	<b>II</b>	Presentation, Chalk & Board
Week 5	static data and static function, <b>Constructors and their types</b> , Destructors, Arrays & Strings: A standard C++ string class.	<b>II</b>	Presentation, Chalk & Board
Week 6	<b>Operator Overloading:</b> Overloading unary and binary operators, Operator Overloading with friend function,	<b>II</b>	<i>Model presentation</i>
Week 7	Data Conversion, type conversion, class to class, basic to class, class to basic	<b>II</b>	Presentation, Chalk & Board, Demonstration
Week 8	Concept of Inheritance, types of inheritance: single, multiple, multilevel, hierarchical, hybrid, protected members,	<b>II</b>	Presentation, Chalk & Board, Demonstration
Week 9	overriding, virtual base class, constructor in derived classes	<b>III</b>	Presentation, Chalk & Board
Week 10	<b>Polymorphism:</b> Pointers in C++, Pointes and Objects, this pointer, virtual and pure virtual functions, implementing polymorphism	<b>III</b>	Presentation, Chalk & Board
Week 11	<b>I/O management:</b> Concept of streams, cin and cout objects, C++ stream classes, Unformatted and formatted I/O, manipulators	<b>IV</b>	Presentation, Chalk & Board

Week 12	<b>File management:</b> File stream, C++ File stream classes, File management functions, File modes, Binary and random files	IV	Presentation, Chalk & Board
Week 13	<b>Object-oriented Design</b> Object modeling using UML, Three models	IV	Presentation, Chalk & Board
Week 14	Class Model (Object and Class Diagram), State model (state Diagram) and Interaction model (Use case diagrams, Activity diagrams, Interaction diagrams).	IV	Presentation, Chalk & Board
Week 15	Revision	IV	Presentation, Chalk & Board

## Subject Mind Mapping



Project