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

Name of Faculty: **Mr.Darshan Solanki**

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: Mr.Darshan Solanki

Department with sitting location: CSE dept, 4th floor Bhanwar Building.

Telephone:

Email: darshansolanki.ce@indusuni.ac.in

Consultation times: **Monday to Friday 4:00 PM to 5:00 PM**

Course lecturer

Full name: 1.Mr.Hiren Mer, 2.Ms.Toral Desai

Department with sitting location: 1.Ground floor Main Building,2.4th floor Bhanwar Building.

Email: hiralsahstri.cse@indusuni.ac.in

Consultation times: **Monday to Friday 4:00 PM to 5:00 PM**

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

Course Objectives

1. To learn the fundamental programming concepts and methodologies which are essential to building good C/C++ programs.
2. To write reusable modules,functions and classes as per Object Oriented Concepts.
3. To enhance employment of students, making good use of the object-oriented programming paradigm to simplify the design and implementation process
4. To encourage the practical problem solving skills.
5. To code, document, test, and implement a well-structured, robust computer program using the C/C++ programming language.

Course Outcomes (CO)

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

1. Understand the difference between the top-down and bottom-up approach.
2. Describe the object-oriented programming approach in connection with C++.
3. Illustrate the process of data file manipulations using C++.
4. Apply the concepts of object-oriented programming.
5. Apply virtual and pure virtual function & complex programming situations.
6. Design and implement C++ programs for complex problems, making good use of the features of the language such as classes, inheritance and templates.

Course Outline

UNIT-I **[12 hours]**

INTRODUCTION TO C++

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.

UNIT-II **[12 hours]**

Objects and classes

Basics of object and class in C++, Private, protected and public Members, static data and static function,

Constructors and their types, Destructors, Arrays & Strings: A standard C++ string class**.**

Operator Overloading: 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 **[12 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++, Pointes 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 **[12 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

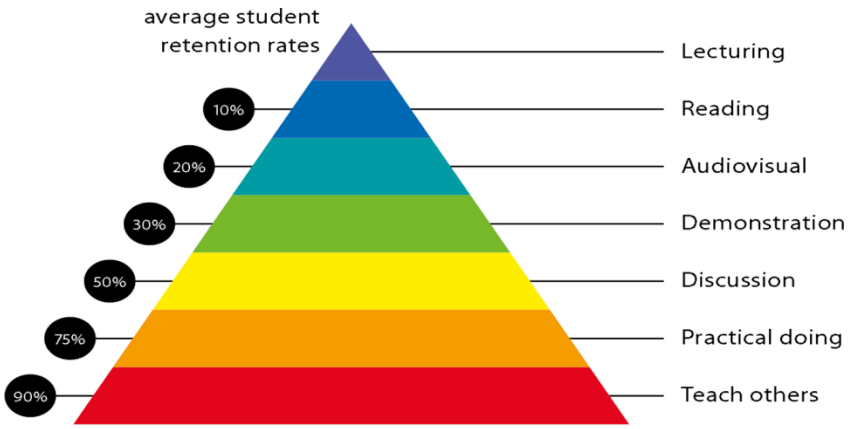

Figure 2: Knowledge retention

Graduate Qualities and Capabilities covered

(Qualities graduates harness crediting this Course)

General Graduate Qualities	Specific Department of _____Graduate Capabilities
Informed 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.	**1 Professional knowledge, grounding & awareness**
Independent learners 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.	**2 Information literacy, gathering & processing**
Problem solvers 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.	**4 Problem solving skills**

Effective communicators 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.	**5 Written communication**
	6 Oral communication
	7 Teamwork
Responsible 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.	**10 Sustainability, societal & environmental impact**

Practical work:

1	Basics of programming	To understand how C++ improves C with object-oriented features.
2	2.1 Write a program to calculate the area of circle, rectangle and square using function overloading. 2.2 Write a program to demonstrate the use of default arguments in function overloading. 2.3 Write a program to demonstrate the use of returning a reference variable.	To learn how to overload functions and operators in C++.
3	3.1 Create a class student which stores the detail about roll no, name, marks of 5 subjects, i.e. science, Mathematics, English, C++. The class must have the following:• Get function to accept value of the data members.• Display function to display values of data members.• Total	To learn how to design C++ classes for code reuse.

	function to add marks of all 5 subjects and store it in the data members named total. 3.2 Create a function power() to raise a number m to power n. the function takes a double value for m and int value for n, and returns the result correctly. Use the default value of 2 for n to make the function calculate squares when this argument is omitted. Write a main that gets the values of m and n from the user to test the function. 3.3 Write a basic program which shows the use of scope resolution operator. 3.4 Write a C++ program to swap the value of private data members from 2 different classes.	
4	4.1 Write a program to illustrate the use of this pointer. 4.2 An election is contested by five candidates. The candidates are numbered 1 to 5 and the voting is done by marking the candidate number on the ballot paper. Write a program to read the ballots and count the votes cast for each candidate using an array variable count. In case a number is read outside the range of 1 to 5, the ballot should be considered as a 'spoilt ballot' and the program should also count the number of spoilt ballots. 4.3 Write a program to call member functions of class in the main function using pointer to object and pointer to member function.	To learn how to design C++ pointers
5	5.1 Using friend function find the maximum number from given two numbers from two different classes. Write all necessary functions and constructors for the program. 5.2 Using a friend function, find the average of three numbers from three different classes. Write all necessary member functions and constructor for the classes. 5.3 Define currency class which contains rupees and paisa as data members. Write a friend function named AddCurrency ( ) which add 2 different Currency objects and returns a Currency object. Write parameterized constructor to initialize the values and use appropriate functions to get the details from the user and display it. 5.4 Create Calendar class with day, month and year as data members. Include default and parameterized constructors to initialize a Calendar object with a valid date value. Define a function AddDays to add days to	To learn how to implement constructors and class member functions.

	the Calendar object. Define a display function to show data in "dd/mm/yyyy" format.	
6	6.1 Create a class named 'String' with one data member of type char *, which stores a string. Include default, parameterized and copy constructor to initialize the data member. Write a program to test this class. 6.2 Write a base class named Employee and derive classes Male employee and Female Employee from it. Every employee has an id, name and a scale of salary. Make a function ComputePay (in hours) to compute the weekly payment of every employee. A male employee is paid on the number of days and hours he works. The female employee gets paid the wages for 40 hours a week, no matter what the actual hours are. Test this program to calculate the pay of employee. 6.3 Create a class called scheme with scheme_id, scheme_name, outgoing_rate, and message charge. Derive customer class form scheme and include cust_id, name and mobile_no data.Define necessary functions to read and display data. Create a menu driven program to read call and message information for a customer and display the detail bill.	To learn how to implement copy constructors and class member functions.
7	7.1 Write a program with use of inheritance: Define a class publisher that stores the name of the title. Derive twoclasses book and tape, which inherit publisher. Book class contains member data called page no and tape class contain time for playing. Define functions in the appropriate classes to get and print the details. 7.2 Create a class account that stores customer name, account no, types of account. From this derive classes cur_acc and sav_acc to include necessary member function to do the following:• Accepts deposit from customer and update balance• Compute and Deposit interest• Permit withdrawal and Update balance. 7.3 Write a base class named Employee and derive classes Male employee and Female Employee from it. Every employee has an id, name and a scale of salary. Make a function ComputePay (in hours) to compute the weekly payment of every employee. A male employee is paid on the number of days and hours he works. The female employee	To learn how containment and inheritance promote code reuse in C++.

	gets paid the wages for 40 hours a week, no matter what the actual hours are. Test this program to calculate the pay of employee.	
8	8.1 Create a class vehicle which stores the vehicleno and chassisno as a member. Define another class for scooter, which inherits the data members of the class vehicle and has a data member for a storing wheels and company. Define another class for which inherits the data member of the class vehicle and hasa data member for storing price and company. Display the data from derived class. Use virtual function. 8.2 Create a base class shape. Use this class to store two double type values that could be used to compute the area of figures. Derive two specific classes called triangle and rectangle from the base shape. Add to the base class, a member function get_data() to initialize the base class data members and another member function display_area() to compute and display the area of figures. Make display_area() as a virtual function and redefine this function in the derived class to suit their requirements. 8.3 Write a program to demonstrate the use of pure virtual function. 8.4 For multiple inheritance, write a program to show the invocation of constructor and destructor. 8.5 Create a class string with character array as a data member and write a program to add two strings with use of operator overloading concept. 8.6 Create a class distance which contains feet and inch as a data member. Overhead = =, <and> operator for the same class. Create necessary functions and constructors too.	To learn how inheritance and virtual functions implement dynamic binding with polymorphism.
9	9.1 Create a class MARIX of size mxn. Overload + and –operators for addition and subtraction of the MATRIX. 9.2 Define a class Coord, which has x and y coordinates as its data members. Overload ++and –operators for the Coord class. Create both its prefix and postfix forms. 9.3 Create one class called Rupees, which has one member data to store amount in rupee and create another class called Paise which has member data to store amount in paise. Write a program to convert one amount to another amount with use of type conversion.	To learn how to overload functions and operators in C++.

	9.4 Create two classes Celsius and Fahrenheit to store temperature in terms of Celsius and Fahrenheit respectively. Include necessary functions to read and display the values. Define conversion mechanism to convert Celsius object to Fahrenheit object and vice versa. Show both types of conversions in main function.	
10	10.1 Write a program to create a function template for finding maximum value contained in an array. 10.2 Write a program to create a class template for the 'Array' class. 10.3 Create a template for the bubble sort function. 10.4 Write a program to illustrate the use of insertion and extraction operators for Text mode Input/Output.	To learn how to design and implement generic classes with C++ templates.
11	11.1 Write a program to illustrate the use of put(), get() and getline() functions for Text mode Input/Output. 11.2 Write a program to illustrate the use of read() and write() functions for Binary mode Input/Output. 11.3 Write a program to illustrate the use of manipulators in file handling.8.Write a program to illustrate the use of file pointer manipulation functions. 11.4 Write down a program to Copy source file 'source.txt' to destination file. 11.5 A file contains a list of telephone numbers in the following format: a) Ram 47890 b) Krishna 878787 c) ---------- d) ---------- The names contain only one word and the names and telephone numbers are separated by white space. Write a Program to read the tel.dat file and display the content. The names should be left justified and the number right-justified.	To learn how to design and implement files with C++.

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:

Theory [Total -100]	Practical [Total -100]
CIE Total :60 **Mid Semester Exam: 40 Marks** **Attendance + Quiz : 10 Marks** **Assignment:10 Marks**	**CIE Total 60** **Internal Practical Exam : 20 marks** **Lab Regularity + Performance : 10 marks** **Practical file :20 Marks** **Open Ended Problem submission: 10 Marks**
ESE total: 40 Marks	**ESE total : 40 Marks**

SUPPLEMENTARY ASSESSMENT

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	I	Presentation, Chalk & Board
	Week 3	Overloading of functions, default arguments, friend functions	I	Presentation, Chalk & Board
	Week 4	**Objects and classes:** Basics of object and class in C++, Private, protected and public Members,	II	Presentation, Chalk & Board
	Week 5	static data and static function, **Constructors and their types**, Destructors, Arrays & Strings: A standard C++ string class**.**	II	Presentation, Chalk & Board
	Week 6	**Operator Overloading**: Overloading unary and binary operators, Operator Overloading with friend function,	II	***Model presentation***
	Week 7	Data Conversion, type conversion, class to class, basic to class, class to basic	II	Presentation, Chalk & Board, Demonstration
	Week 8	Concept of Inheritance, types of inheritance: single, multiple, multilevel, hierarchical, hybrid, protected members,	II	Presentation, Chalk & Board, Demonstration

	Week 9	overriding, virtual base class, constructor in derived classes	***III,V***	Presentation, Chalk & Board
	Week 10	**Polymorphism:** Pointers in C++, Pointes and Objects, this pointer, virtual and pure virtual functions, implementing polymorphism	***III,V***	Presentation, Chalk & Board
	Week 11	**I/O management**: Concept of streams, cin and cout objects, C++ stream classes, Unformatted and formatted I/O, manipulators	***IV,VI***	Presentation, Chalk & Board
	Week 12	**File management**: File stream, C++ File stream classes, File management functions, File modes, Binary and random files	III,V	Presentation, Chalk & Board
	Week 13	**Object-oriented Design** 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

COMPUTER ENGINEERING DEPARTMENT COURSE DEPENDANCY CHART