Course Code **CSE2001**	**Object Oriented Programming with C++**	**Course Type** **LTP**	**Credits** **4**

Course Objectives:

- To understand the principles of Object oriented programming
- To identify and practice the object oriented programming concepts and techniques
- To solve a real world problems through object oriented approach

Course Outcomes:

At the end of the course, students should able to

- familiarize object oriented paradigm
- apprehend the syntax and semantics of the C++ programming language
- reuse the code
- adopt object oriented design approach to solve a real world problems
- Synthesize generic class templates

Student Outcomes (SO): a,b,c,l

a. An ability to apply the knowledge of mathematics, science and computing appropriate to the discipline

b. An ability to analyze a problem, identify and define the computing requirements appropriate to its solution.

c. An ability to design, implement and evaluate a system / computer-based system, process, component or program to meet desired needs

l. An ability to apply mathematical foundations, algorithmic principles and computer science theory in the modelling and design of computer-based systems (CS)

Unit No	**Unit Content**	**No. of hours**	**SOs**
1	**Introduction to object oriented approach**: Why object oriented programming?- Characteristics of object oriented language: classes and objects - encapsulation-data abstraction- inheritance - polymorphism - Merits and Demerits of object oriented programming. UML- class diagram of OOP - Inline function – default argument function- Exception handling(Standard) - reference: independent reference – function returning reference – pass by reference	**6**	**a,b,c**
2	**Classes and objects:** Definition of classes – access specifier – class versus structure – constructor – destructor – copy constructor and its importance – array of objects – dynamic objects- friend function-friend class – container class	**10**	**a,b,c**

3	**Polymorphism and Inheritance :** Polymorphism-compile time polymorphism – function overloading – operator overloading - . Inheritance-types of inheritance- constructors and destructors in inheritance – constraints of multiple inheritance-Abstract base class – pure virtual functions- run time polymorphism-function overriding.	**10**	**a,b,c**
4	**Exception handling and Templates** Exception handling(user-defined exception)- Function template , Class template – Template with inheritance , STL – Container, Algorithm, Iterator -vector, list, stack, map	**10**	**a,b,c**
5	**IOstreams and Files** IOstreams, Manipulators- overloading Inserters(<<) and Extractors(>>)-Sequential and Random files – writing and reading objects into/from files – binary files	**7**	**a,b,c**
6	**Guest Lecture on Contemporary Topics**	**2**	
	Total Hours:	**45**	

Mode of Teaching and Learning*: Flipped Class Room, Activity Based Teaching/Learning, Digital/Computer based models, wherever possible to augment lecture for practice/tutorial and minimum 2 hours lectures by industry experts on contemporary topics*

Mode of Evaluation and assessment:

The assessment and evaluation components may consist of unannounced open book examinations, quizzes, student's portfolio generation and assessment, and any other innovative assessment practices followed by faculty, in addition to the Continuous Assessment Tests and Term End Examinations.

Reference Books:

1.	Stanley B Lippman, Josee Lajoie, Barbara E, Moo, “C++ primer”, Fifth edition, Addison-Wesley, 2012
2.	Bjarne Stroustrup, The C++ programming Language, Addison Wesley, 4th edition, 2013
3.	Harvey M. Deitel and Paul J. Deitel, C++ How to Program, 7th edition, Prentice Hall, 2010
4.	Maureen Sprankle and Jim Hubbard, Problem solving and Programming concepts, 9th edition, Pearson Eduction, 2014

Indicative list of Experiments

No.	Description of Experiments	SO
1	Design a class to represent a bank account. Include the following members. Data Members • Name of the depositor • Account number • Type of account • Balance amount in the account Methods • To assign initial values • To deposit an amount • To withdraw an amount after checking balance • To display the name and balance Incorporate a constructor to provide initial values.	**I**
2	Guess-the-number-game: Write a program that pays the game of “guess the number” as follows: Your program choose the number to be guess by selecting an integer at random in the range 1 to 1000. The program then displays the following: I have a number between 1 and 1000. Can you guess my number? Please type your first guess. The player then type a first guess. The program responds with one of the following: 1. Excellent ! you guessed the number! Would like to play again (y or n)? 2. Too low. Try again. 3. Too high. Try again. If the payer’s guess is incorrect, your program should loop until the player finally get the number right. Your program should keep telling the player Too high or Too low to help the player “zero in” on the correct answer.	
3	Assume that a bank maintains two kinds of account for its customers, one called saving account and the other current account. The saving account provides compound interest and withdrawal facilities but no cheque book facility. The current account provides cheque book facility but no interest. Current account holders should also maintain a minimum balance falls below this level, a service charge is imposed. Create a class Account that stores customer name, account number, and	**I**

	type of account. From this device the classes Curr-acct and Sav-acct to make them more specific to their requirements. Include the necessary methods in order to achieve the following tasks. • Accept deposit from a customer and update the balance • Display the balance • Compute and deposit interest • Permit withdrawal and update the balance • Check for the minimum balance, impose penalty, if necessary and update the balance. Do not use any constructors. Use methods to initialize the class members.	
4	An election is contested by 5 candidates. The candidates are numbered 1 to 5 and the voting is done by marking the candidate number on the ballot paper. Write a C++ program to read the ballots and count the votes cast for each candidate using an array. In case, a number read is outside the range 1 to 5, the ballot should be considered as a ‘spoilt ballot’ and the program should also count the number of spoilt ballots.	**l**
5	Develop a program which will read a string and rewrite it in the alphabetical order. For example, the word STRING should be written as GINRST	**l**
6	Create a class by name date with the member data day, month and year. Perform the following: • Overload all relational operators <,<=,>,>=,==,!= • Overload ++ operator to increment a date by one day • Overload + to add given number of days to find the next date • Provide the necessary function to use the statement like days=dt; where days is an int variable and dt is an object of date class. The statement is intended to assign the number of days elapsed in the current year of the date to the variable days. Note that this is a case of conversion from derived type to basic type.	**l**
7	Develop a program to sort a file consisting of books’ details in the alphabetical order of author names. The details of books include book_id, author_name, price, no_of_pages, publisher, year_of_publishing	**l**
8	Design a class template by name Vector and perform the following: • Find the smallest of the element in the Vector • Search for an element in the Vector • Find the average of the element in the array	**l**
9	Design a generic function for finding the largest of three numbers.	**l**