Course Objectives: -

- Articulate the principles of object-oriented problem solving and programming.
- Outline the essential features and elements of the C++ programming language.
- Explain programming fundamentals, including statement and control flow and recursion.
- Apply the concepts of class, method, constructor, instance, data abstraction, function abstraction, inheritance, overriding, overloading, and polymorphism.
- Program with basic data structures using array.
- Program using objects and data abstraction, class, and methods in function abstraction.
- Analyze, write, debug, and test basic C++ codes using the approaches introduced in the course.
- Analyze problems and implement simple C++ applications using an object-oriented software engineering approach.

Learning Outcomes: -

- Programming fundamentals, including statement and control flow and recursion.
- Learn concepts of class, method, constructor, instance, data abstraction, function abstraction, inheritance, overriding, overloading, and polymorphism.
- Fundamentals of basic data structures using array
- To learn about objects and data abstraction, class, and methods in function abstraction.

Unit Wise Mapping: -

Unit	Unit Title	Skill	Employability	Entrepreneurship
1	**Introduction to dynamic memory allocation**	YES		
2	**Introduction to OOP**	YES		
3	**C++ Programming basics**	YES		
4	**Object and Classes**	YES		
5	**Arrays and string arrays fundamentals**	YES		
6	**Forms of Polymorphism**	YES	YES	
7	**Inheritance**	YES	YES	
8	**Pointer**	YES	YES	
9	**Virtual Function:**	YES	YES	
10	**I/O Streams and Files**	YES	YES	YES
11	**Templates, STL and Exceptions**	YES	YES	YES

	GANPAT UNIVERSITY							
	INSTITUTE OF COMPUTER TECHNOLOGY							
Programme	Bachelor of Technology				Branch/Spec.	Computer Science & Engineering (CBA/CS/BDA)		
Semester	II				Version	1.0.1.1		
Effective from Academic Year		2021-22			Effective for the batch Admitted in			June 2021
Subject code	2CSE203		Subject Name		ESSENTIALS OF SOFTWARE FOUNDATION & PROGRAMMING – II			

Teaching scheme					Examination scheme (Marks)			
(Per week)	Lecture(DT)		Practical(Lab.)		Total		CE	SEE	Total
	L	TU	P	TW				
Credit	3	0	2	0	5	Theory	40	60	100
Hours	3	0	4	0	7	Practical	60	40	100

Pre-requisites:
C Programming, Flow Charting.

Learning Outcome:
After completing the Course, students will learn: ● Programming fundamentals, including statement and control flow and recursion. ● Learn concepts of class, method, constructor, instance, data abstraction, function abstraction, inheritance, overriding, overloading, and polymorphism. ● Fundamentals of basic data structures using array ● To learn about objects and data abstraction, class, and methods in function abstraction.

Theory syllabus		
Unit	Content	Hours
1	**Introduction to dynamic memory allocation** C programming basics, dynamic memory allocation functions	3
2	**Introduction to OOP** What is object-oriented programming? Why do we need object oriented, Programming characteristics of object-oriented languages, C and C++.	3
3	**C++ Programming basics** Output using cout Directives. Input with cin, type bool,the setw manipulator, type conversions.	3
4	**Object and Classes** Making sense of core object concepts (Encapsulation, Abstraction, Polymorphism, Classes, Messages Association, Interfaces) Implementation of class in C++, C++ Objects as physical object, C++ object as data types constructor. Object as function arguments. The default copy constructor, returning object from function. Structures and classes. Classes objects and memory static class data. Const and classes	5
5	**Arrays and string arrays fundamentals** Arrays as class member data, Arrays of object, string, the standard C++ String class	4
6	**Forms of Polymorphism** Overloading unary operations. Overloading binary operators, data conversion, pitfalls of operators overloading and conversion keywords, inline function, function overloading, Explicit and Mutable.	5
7	**Inheritance** Concept of inheritance. Derived class and based class. Derived class constructors, member function, inheritance in the English distance class, class hierarchies, inheritance and graphics shapes, public and private inheritance, aggregation: Classes within classes, inheritance and program development	4

8	**Pointer**	5
	Addresses and pointers. The address of operator and pointer and arrays. Pointer and Faction pointer and C-types string. Memory management -New and Delete, pointers to objects, debugging pointers	
9	**Virtual Function:**	4
	Virtual Function, friend function, Static function, Assignment and copy initialization, this pointer, dynamic type information	
10	**I/O Streams and Files**	4
	Streams classes, Stream Errors, Disk File I/O with streams, file pointers, error handling in file I/O with member function, overloading the extraction and insertion operators, memory as a stream object, command line arguments, and printer output	
11	**Templates,STL and Exceptions**	5
	Function templates and class templates, Introduction to exception, try-catch-throw, multiple catch, catch all, rethrowing exception, implementing user defined exceptions, Overview and use of Standard Template Library	

Practical content

- Implement programs to learn dynamic memory allocation using its functions.
- Implement programs with the use of cin, cout and various arithmetic operators.
- Write programs implementing signed & unsigned types, other data types, type conversion and library functions.
- Implement the programs using manipulators, various operators, looping and decision structures.
- Implement programs of class, objects and array concepts.
- Implement programs to learn about string objects, enum types and operator overloading
- Implement programs of single inheritance, hierarchical inheritance, multiple and multilevel inheritance, hybrid inheritance and aggregation.
- Write programs implementing pointers, polymorphism and virtual functions.
- Write programs implementing friend function, static member function, this pointer, operator overloading with file and command line argument.
- Implement programs to learn about reading and writing files with OOP.
  Implement programs to learn about function template, class template and exception handling.

NPTEL Course: Programming in C++, Spoken tutorial test of basics of c & C++

https://nptel.ac.in/courses/106/105/106105151/

Text Books

1	Object Oriented Programming in C++ By Robert Lafore

Reference Books

1	C++ Programming, Black Book, Steven Holzner, dreamtech
2	C++ : The complete Reference By Herbert Schildt
3	Object Oriented Programming with C++ By E. Balagurusamy