A. Course Handout (Version 1.0) | Last updated on 5th March, 2022

Institute Name	Chitkara University Institute of Engineering & Technology		
Department Name	Department of Computer Science & Engineering		
Programme Name	Bachelor of Engineering (B.E.) - Computer Science & Engineering		
Course Name	OOPs using C++	Session	2021-2022
Course Code	CS102	Semester/Batch	2nd/ 2021
L-T-P (Per Week)	2-0-4	Course Credits	04
Course Coordinator Name	Dr. Anshu Singla		

1. Scope and Objectives of the Course

The main scope of the course is to develop small-scale C++ programs using programming constructs of data types, arrays, pointers, and object-oriented approaches. The main scope and objectives of the course are:

- To formulate new solutions of programming problems using object oriented approach of programming.
- To demonstrate the importance of major features of OOP such as encapsulation, inheritance, code extensibility, reusability, and polymorphism.
- To customize their own Templates and implement the generic programming.
- To evaluate and improve the existing programs using Standard Template Library.

2. Course Learning Outcomes

After completion of the course, students will be able to do the following:

CLO1: Analyze the problem statement.

CLO2: Choose the appropriate OOPs programming constructs to solve the problems.

CLO3: Differentiate between efficient and inefficient way of programming.

CLO4: Determine the bugs in a program and recognize needed basic operations.

CLO5: Formulate new solutions for programming problems or improve existing code to program effectively.

CLO-PO mapping grid |Program outcomes (Pos) are available as a part of Academic Program Guide (APG)

Course Learning Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CLO1		H	H		M						M	
CLO2		H	H	M								
CLO3					H							
CLO4		H	H						M		H	

CL05		H	H		H							

3. Recommended Books (Reference Books/Text Books)

RB1: 'Object Oriented Programming with C++' by E Balagurusamy, 6th Edition, Tata McGraw Hill.

RB2: Object Oriented Programming in C++' by Robert Lafore, 4th Edition, Galgotia.

RB3: The Complete Reference C++' by Herbert Schildt , 4th Edition ,Tata McGraw Hill.

RB4: Stroustrup, Bjarne, The C++ Programming Language, Pearson Education .

RB5: Lippman, S.B. and Lajoie, J., C++Primer, Pearson Education .

4. Other readings and relevant websites:

Serial No	Link of Journals, Magazines, Websites and Research Papers
1.	http://www.cprogramming.com/tutorial/c++-tutorial.html
2.	http://www.cplusplus.com/doc/tutorial/
3.	http://www.tenouk.com/cncplusplustutorials.html
4.	http://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-088-introduction-to-c-memory-management-and-c-object-oriented-programming-january-iap-2010/

5. Recommended Tools and Platforms

- Coding Blocks, Dev-C++
- Coding Ninjas (online platform- https://codingninjas.com/)

6. Course Plan:

Lecture Number	Topics
1-2	Introduction to basic concepts of object-oriented programming, Procedural Programming versus Object Oriented Programming paradigm, Structures versus Classes, Variables and Data Types in C++, Console I/O, Preprocessor directives
3	Decision making (if, if-else, if- else-if ladder, switch case, go to), Repetitive Constructs (for, while, do-while, break, continue)
4-6	Arrays (1D, 2D, multi dimensional), Understanding pointers, Accessing address of a variable, Declaring & initializing pointers, Accessing a variable through its pointer, Pointer arithmetic, Pointer to a pointer, Accessing arrays using pointer
7-8	Functions in C++: Inline functions, Default arguments, Function prototyping, Function overloading, Call by Reference, Call by Value & Call by pointer, Return by Reference.
9-10	Classes and Objects: OOps Paradigm and its features, Specifying a class, Creating class objects, Accessing class members, Access specifiers – public,private,and protected, Objects and memory, Static members, Static objects, Constant member function, Constant objects, Friend functions, Friend class, Passing Object as an argument (by value, by reference and by address), Returning object from a function.
11	Constructors and Destructors: Need for constructors and destructors, , Constructor overloading, Copy constructor, Dynamic constructors, Destructors, Constructors and destructors with static members.

12-13	Operator Overloading: Defining operator overloading, Rules for overloading operators, Overloading of unary operators, Binary operators(+,-,/,*), Binary operators using friend functions, Manipulation of strings using operators Overloading(>,<,= =) , Type conversion: Basic type to class type, Class type to basic type, Class to class type.
14-15	Dynamic memory management & Pointers - new and delete Operators, Pointers and classes, Pointer to an object, Pointer to a member, this Pointer, Possible problems with the use of pointers - Dangling/wild pointers, Null pointer assignment, Memory leak and allocation failures.
16-17	Inheritance: Introduction, Defining derived classes, Forms of inheritance (single, multilevel, multiple, hybrid & hierarchical), Ambiguity in multiple and multipath inheritance
18	Inheritance with constructor, Overriding member functions, Order of execution of constructors and destructors.
19	Concept of Binding, Early binding and late binding, Virtual functions, Pure virtual functions, Abstract classes, Virtual destructors & polymorphism
20	Exception Handling: Review of traditional error handling, Basics of exception handling, Exception handling mechanism, Throwing mechanism, Catching mechanism, Rethrowing an exception, Specifying exceptions.
21	Function templates, Class templates, Overloading of template functions.

22	Introduction to the Standard Template Library (Containers, Algorithms, Iterators) :
	Sequence Container: vector(push_back(),pop_back(),back(),size(),empty()),
	list(push_front(),pop_front(),front(),size(),empty())
	dequeue(push_back(),pop_back(),push_front(),pop_front(),size(),empty())

23	Associative Container: set(Insert(),erase(),Size(),Empty(),Count(),Clear()), multiset(Insert(),erase(),Size(),Empty(),Count(),Clear()), map(Insert(),erase(),Size(),Empty(),Count(),Clear()), multimap(Insert(),erase(),Size(),Empty(),Count(),Clear())
24	Derived Container: stack, queue, priority_queue, ALGORITHMS: count(),count_if(),find(),find_if(),copy(),fill(),remove(), remove_copy(),replace(),replace_copy(),reverse(),reverse_copy(), unique(),unique_copy(),max(),max_element(),min(),min_element(), iterators: input, output, forward, vectors:back(),begin(),clear(),empty(),end(),erase(),pop_back(),push_back()

7. Lab Plan

Sr. No.	Lab Number	Experiment	Platform	Learning Resource
1	1-4	Console input/output cin, cout; Structure of C++ program; Preprocessor directives; Different Data Types and its size; Constants	C++	• https://www.programiz.com/cpp-programming/input-output • https://www.tutorialspoint.com/cplusplus/cpp_basic_input_output.htm
2	3-8	Programs to demonstrate decision	C++	• https://www.tutorialspoint.com/cplusplus/cpp_loop_types.htm

		making, looping constructs. Menu driven programs wil real life scenario like banking application.		• https://www.javatpoint.com/cpp-if-else
3	9-14	Linear Search, Binary Search, Selection Sort, Insertion Sort, Bubble Sort, Programs using arrays. Matrix Operations, Pointer Arithmetic	C++	• https://www.tutorialspoint.com/data_structures_algorithms/binary_search_algorithm.htm • https://www.studytonight.com/data-structures/bubble-sort • https://www.tutorialspoint.com/cprogramming/c_pointer_arithmetic.htm
4	15-16	Functions using call by reference and call by value	C++	• https://www.geeksforgeeks.org/functions-in-c/?ref=lbp
5	17-22	OOPs cocepts showing run-time entity, difference between class, object with real scenario like Student-course enrolment in school, Inventory system Usage of constructor, destructor	C++	• https://www.geeksforgeeks.org/c-classes-and-objects/?ref=lbp • https://www.javatpoint.com/cpp-oops-concepts • https://www.geeksforgeeks.org/constructors-c/?ref=lbp
6	23-24	this pointer, pointer to object, Pointer to member, Null pointer , Dangling pointer	C++	• https://www.javatpoint.com/cpp-pointers
7	25-28	Single level Inheritance, Multilevel inheritance, Method Overloading and Overriding. Access specifier-private, protected, public usage	C++	• https://www.javatpoint.com/cpp-inheritance
8	29-32	Run time polymorphism , Behaviour of constructors,its overloading, and its order of execution in inheritance with real life example like inventory system	C++	• https://www.javatpoint.com/cpp-polymorphism
9	33-36	Abstract classes, Virtual functions demonstration taking	C++	• https://www.simplilearn.com/tutorials/cpp-tutorial/abstract-class-in-cpp

		real life example like Student Admission process		
10	37-38	Exception Handling covering types of exceptions like Arithmetic exception, Input/Output Exception, Throw and Catch mechanism, Rethrow exceptions	C++	• https://www.geeksforgeeks.org/exception-handling-c/?ref=lbp
11	39-42	Function templates, Class templates, Overloading of template functions.	C++	• https://www.geeksforgeeks.org/templates-cpp/ • https://www.w3schools.in/cplusplus-tutorial/templates/
12	43-44	Sequence Container: vector(push_back(),pop_back(),back(),size(),empty()), list(push_front(),pop_front(),front(),size(),empty()) dequeue(push_back(),pop_back(),push_front(),pop_front(),size(),empty())	C++	• https://www.javatpoint.com/cpp-stl-components • https://www.javatpoint.com/cpp-iterators
13	45-46	Associative Container: set(Insert(),erase(),Size(),Empty(),Count(),Clear()), multiset(Insert(),erase(),Size(),Empty(),Count(),Clear()), map(Insert(),erase(),Size(),Empty(),Count(),Clear()), multimap(Insert(),erase(),Size(),Empty(),Count(),Clear())	C++	• https://www.codeguru.com/cplusplus/an-introduction-to-ordered-associative-containers-in-c/
14	47-48	Derived Container: stack, queue, priority_queue, ALGORITHMS: count(),count_if(),find(),find_if(),copy(),fill(),remove(), remove_copy(),replace(),replace_copy(),reverse(),reverse_copy(), unique(),unique_copy(),max(),max_eleme	C++	• https://www.geeksforgeeks.org/priority-queue-in-cpp-stl/?ref=lbp

		nt(),min(),min_element(), iterators: input, output, forward, vectors:back(),begin(),clear(),empty(),end(),erase(),pop_back(),push_back()		

8. Delivery/Instructional Resources

Lecture Number	Topics	PPT *(Link of ppts on the central server)*	Industry Expert Session *(If **yes**: link of ppts on the central server)*	Web References	Audio-Video
1-2	Introduction to basic concepts of object-oriented programming, Procedural Programming versus Object Oriented Programming paradigm, Structures versus Classes, Variables and Data Types in C++, Console I/O, Preprocessor directives			• https://www.cet.edu.in/noticefiles/285_OOPS%20lecture%20notes%20Complete.pdf • https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-096-introduction-to-c-january-iap-2011/lecture-notes/ • https://www.cs.bham.ac.uk/~jxb/DSA/dsa.pdf	• https://onlinecourses.nptel.ac.in/noc19_cs39/preview • https://www.youtube.com/watch?v=jVJwDy1zUUg • https://www.youtube.com/watch?v=LZFoktwiars&list=PL0gIV7t6l2iIsR55zsSgeiOw9Bd_IUTbY • https://www.youtube.com/watch?v=zWg7U0OEAoE&list=PLBF3763AF2E1C572F
3	Decision making (if, if-else, if- else-if ladder, switch case, go to), Repetitive Constructs (for, while, do-while, break, continue)			• https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-096-introduction-to-c-january-iap-2011/lecture-notes/	• https://www.youtube.com/watch?v=efXI8anQwXo&list=PLEAYkSg4uSQ2qzihjdDEseWrrY1DyxH9P
4-6	Arrays (1D, 2D, multi dimensional), Understanding pointers, Accessing address of a variable, Declaring			• https://www.hackerearth.com/practice/algorithms/sorting/bubble-sort/visualize/	• https://www.youtube.com/watch?v=Z_0xXmOgYtY • https://www.youtube.co

	& initializing pointers, Accessing a variable through its pointer, Pointer arithmetic, Pointer to a pointer, Accessing arrays using pointer			• https://cse.iitrpr.ac.in/ckn/courses/f2015/csl201/w1.pdf • https://www.cs.bham.ac.uk/~jxb/DSA/dsa.pdf • https://www3.ntu.edu.sg/home/ehchua/programming/cpp/cp4_PointerReference.html • https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-096-introduction-to-c-january-iap-2011/lecture-notes/	m/watch?v=nAGjoysNM4s
7-8	Functions in C++: Inline functions, Default arguments, Function prototyping, Function overloading, Call by Reference, Call by Value & Call by pointer, Return by Reference.			• https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-096-introduction-to-c-january-iap-2011/lecture-notes/	• https://www.youtube.com/watch?v=efXI8anQwXo&list=PLEAYkSg4uSQ2qzihjdDEseWrrY1DyxH9P
9-10	Classes and Objects: OOps Paradigm and its features, Specifying a class, Creating class objects, Accessing class members, Access specifiers – public,private,and protected, Objects and memory, Static members, Static objects, Constant member function, Constant objects, Friend functions, Friend class, Passing Object as an			• https://www.cse.iitb.ac.in/~cs101/2019.1/lectures/Lecture20.pdf • https://www.cet.edu.in/noticefiles/285_OOPS%20lecture%20notes%20Complete.pdf • https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-096-	• https://www.youtube.com/watch?v=i_5pvt7ag7E

	argument (by value, by reference and by address), Returning object from a function.			introduction-to-c-january-iap-2011/lecture-notes/	
11	Constructors and Destructors: Need for constructors and destructors, constructor overloading, Copy constructor, Dynamic constructors, Destructors, Constructors and destructors with static members.			• https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-096-introduction-to-c-january-iap-2011/lecture-notes	• https://www.youtube.com/watch?v=uIjgl6qBfFc • https://www.youtube.com/watch?v=oRBK0Mh_gG0
12-13	Operator Overloading: Defining operator overloading, Rules for overloading operators, Overloading of unary operators, Binary operators(+,-,/,*), Binary operators using friend functions, Manipulation of strings using operators Overloading(>,<,= =), Type conversion: Basic type to class type, Class type to basic type, Class to class type.			• https://www3.ntu.edu.sg/home/ehchua/programming/cpp/cp4_PointerReference.html • https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-096-introduction-to-c-january-iap-2011/lecture-notes/	• https://www.youtube.com/watch?v=Z_0xXmOgYtY • https://www.youtube.com/watch?v=nAGjoysNM4s
14-15	Dynamic memory management & Pointers - new and delete Operators, Pointers and classes, Pointer to an object, Pointer to a member, this Pointer, Possible problems with the use of pointers - Dangling/wild pointers, Null pointer assignment, Memory leak and allocation failures.			• https://www3.ntu.edu.sg/home/ehchua/programming/cpp/cp4_PointerReference.html • https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-096-introduction-to-c-january-iap-2011/lecture-	• https://www.youtube.com/watch?v=Z_0xXmOgYtY • https://www.youtube.com/watch?v=nAGjoysNM4s

				notes/	
16-17	Inheritance: Introduction, Defining derived classes, Forms of inheritance (single, multilevel, multiple, hybrid & hierarchical), Ambiguity in multiple and multipath inheritance			• https://view.officeapps.live.com/op/view.aspx?src=https%3A%2F%2Fjpwebdevelopers.in%2Fppts%2Finheritance.pptx&wdOrigin=BROWSELINK	• https://www.youtube.com/watch?v=8fDao3MBbwk
18	Inheritance with constructor, Overriding member functions, Order of execution of constructors and destructors.			• https://www.w3schools.in/cplusplus-tutorial/inheritance/	• https://www.youtube.com/watch?v=jflvIa60EAg
19	Concept of Binding, Early binding and late binding, Virtual functions, Pure virtual functions, Abstract classes, Virtual destructors & polymorphism			• https://www.w3schools.in/cplusplus-tutorial/polymorphism/	• https://www.youtube.com/watch?v=jflvIa60EAg
20	Exception Handling: Review of traditional error handling, Basics of exception handling, Exception handling mechanism, Throwing mechanism, Catching mechanism, Rethrowing an exception, Specifying exceptions.			• https://www.coursehero.com/file/129688644/Lecture-9-Exception-handlingppt/	• https://www.youtube.com/watch?v=-frZ8btNBSU
21	Function templates, Class templates, Overloading of template functions.			• https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-096-introduction-to-c-january-iap-2011/lecture-	• https://www.youtube.com/watch?v=zjXso3X2jms • https://www.youtube.com/watch?v=zjXso3X2jms

				notes/MIT6_096IAP11_lec09.pdf • https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-096-introduction-to-c-january-iap-2011/lecture-notes/MIT6_096IAP11_lec09.pdf	
22	Introduction to the Standard Template Library (Containers, Algorithms, Iterators) : Sequence Container: vector(push_back(), pop_back(),back(),size(),empty()), list(push_front(),pop_front(),front(),size(),empty()) dequeue(push_back(),pop_back(),push_front(),pop_front(),size(),empty())			• https://www.geeksforgeeks.org/cpp-stl-tutorial/ • https://courses.cs.washington.edu/courses/cse333/18su/lectures/14-c++-STL.pdf	• https://www.youtube.com/watch?v=W7uB9-TKfTg
23	Associative Container: set(Insert(),erase(),Size(),Empty(),Count(),Clear()), multiset(Insert(),erase(),Size(),Empty(),Count(),Clear()), map(Insert(),erase(),Size(),Empty(),Count(),Clear()), multimap(Insert(),erase(),Size(),Empty(),Count(),Clear())			• https://www.geeksforgeeks.org/cpp-stl-tutorial/	• https://www.youtube.com/watch?v=W7uB9-TKfTg
24	Derived Container: stack, queue, priority_queue, ALGORITHMS: count(),count_if(),find(),find_if(),copy(), fill(),remove(), remove_copy(),replace(),replace_copy(),reverse(),reverse_c			• https://www.geeksforgeeks.org/cpp-stl-tutorial/	• https://www.youtube.com/watch?v=W7uB9-TKfTg

	opy(), unique(),unique_copy(),max(),max_element(),min(),min_element(), iterators: input, output, forward, vectors:back(),begin(),clear(),empty(),end(),erase(),pop_back(),push_back()				

9. Action plan for different types of learners

Slow Learners	Average Learners	Fast Learners
• Remedial Classes on Saturdays • Encouragement for improvement using Peer Tutoring	• Formative Exercises used to highlight concepts and notions • Pre-coded algorithms to illustrate concepts and notions • E-notes and E-exercises to read in addition to pedagogic material.	• Design solutions for complex problems • Presentation on topics beyond those covered in CHO • Engaging students to hold hands of slow learners by creating a Peer Tutoring Group • Participation in Hackathons, competitions.

10. Evaluation Scheme & Components:

Evaluation Component	Type of Component	No. of Assessments	Weightage of Component	Mode of Assessment
Component 1	Formative Assessments (FAs)	03*	20%	Online
Component 2	Subjective Test/Sessional Tests (STs)	03**	30%	Online
Component 3	End Term Examinations	01	50%	Online
Total		**100%**		

*Out of 03FAs, the ERP system will automatically pick marks of the best 02 FAs for final marks evaluation of FAs.

** Out of 03STs, the ERP system will automatically pick marks of the best 02 STs for final marks evaluation of STs.

11. Details of Evaluation Components:

Evaluation Component	Description	Syllabus Covered (%)	Timeline of Examination	Weightage (%)
Component 01	FA1	Up to 25%	25th March, 2022	20%

	FA2	26%-55%	29th April, 2022	
	FA3	55%-85%	23rd May, 2022	
Component 02	ST 01	Up to 30%	4th April, 2022	30%
	ST 02	31% - 70%	9th May, 2022	
	ST 03	71% - 100%	6th June, 2022	
Component 03	End Term Examination*	100%		50%
Total				100%

*As per Academic Guidelines minimum of 75% attendance is required to become eligible for appearing in the End Semester Examination

Evaluation Components

Type of Assessment	Timeline of Conduct	Total Marks	Question Paper Format			
			1 Mark MCQ	2 Mark MCQ	5 Mark Algorithm/Coding	10 Mark Algorithm/Coding
Formative Assessment 1	25th March, 2022	20	15	-	1	-
Sessional Test 1	4th April, 2022	30	5	5	1	1
Formative Assessment 2	29th April, 2022	20	15	-	1	-
Sessional Test 2	9th May, 2022	30	5	5	1	1
Formative Assessment 3	23rd May, 2022	20	15	-	1	-
Sessional Test 3	6th June, 2022	30	5	5	1	1
End Term Examination		50	10	5	4	1

12. Syllabus of the Course:

Lecture Number	Topics	No. of Lectures	Weightage %
1-2	Introduction to basic concepts of object-oriented programming, Procedural Programming versus Object Oriented Programming paradigm, Structures versus Classes, Variables and Data Types in C++, Console I/O, Preprocessor directives.	2	10%
3	Decision making (if, if-else, if- else-if ladder, switch case, go to), Repetitive Constructs (for, while, do-while, break, continue).	8	20%
4-6	Arrays (1D, 2D, multi dimensional), Understanding pointers, Accessing address of a variable, Declaring & initializing pointers, Accessing a variable through its pointer, Pointer arithmetic, Pointer to a pointer, Accessing arrays using pointer.		
7-8	Functions in C++: Inline functions, Default arguments, Function prototyping, Function overloading, Call by Reference, Call by Value & Call by pointer, Return by Reference.		
9-10	Classes and Objects: OOps Paradigm and its features, Specifying a class,		

	Creating class objects, Accessing class members, Access specifiers – public,private,and protected, Objects and memory, Static members, Static objects, Constant member function, Constant objects, Friend functions, Friend class, Passing Object as an argument (by value, by reference and by address), Returning object from a function.		
11	Constructors and Destructors: Need for constructors and destructors, Constructor overloading, Copy constructor, Dynamic constructors, Destructors, Constructors and destructors with static members.	8	30%
12-13	Operator Overloading: Defining operator overloading, Rules for overloading operators, Overloading of unary operators, Binary operators(+,-,/,*), Binary operators using friend functions, Manipulation of strings using operators Overloading(>,<,= =) , Type conversion: Basic type to class type, Class type to basic type, Class to class type.		
14-15	Dynamic memory management & Pointers - new and delete Operators, Pointers and classes, Pointer to an object, Pointer to a member, this Pointer, Possible problems with the use of pointers - Dangling/wild pointers, Null pointer assignment, Memory leak and allocation failures.		
16-18	Inheritance: Introduction, Defining derived classes, Forms of inheritance (single, multilevel, multiple, hybrid & hierarchical), Ambiguity in multiple and multipath inheritance. Inheritance with constructor, Overriding member functions, Order of execution of constructors and destructors.		
19	Concept of Binding, Early binding and late binding, Virtual functions, Pure virtual functions, Abstract classes, Virtual destructors & polymorphism	3	20%
20	Exception Handling: Review of traditional error handling, Basics of exception handling, Exception handling mechanism, Throwing mechanism, Catching mechanism, Rethrowing an exception, Specifying exceptions.		
21	Function templates, Class templates, Overloading of template functions.		
22	Introduction to the Standard Template Library (Containers, Algorithms, Iterators) : Sequence Container: vector(push_back(),pop_back(),back(),size(),empty()), list(push_front(),pop_front(),front(),size(),empty()) dequeue(push_back(),pop_back(),push_front(),pop_front(),size(),empty())	3	20%
23	Associative Container: set(Insert(),erase(),Size(),Empty(),Count(),Clear()), multiset(Insert(),erase(),Size(),Empty(),Count(),Clear()), map(Insert(),erase(),Size(),Empty(),Count(),Clear()), multimap(Insert(),erase(),Size(),Empty(),Count(),Clear())		
24	Derived Container: stack, queue, priority_queue, ALGORITHMS: count(),count_if(),find(),find_if(),copy(),fill(),remove(), remove_copy(),replace(),replace_copy(),reverse(),reverse_copy(), unique(),unique_copy(),max(),max_element(),min(),min_element(), iterators: input, output, forward, vectors:back(),begin(),clear(),empty(),end(),erase(),pop_back(),push_back()		

This Document is designed and approved by:

Designation	**Name**	**Signature**
Course Coordinator	Dr. Anshu Singla	
Head Academic Delivery	Dr. Vikas Khullar	
Cluster Dean	Dr. Rishu Chhabra	
Dean (Academic Affairs)	Dr. Rajnish Sharma	
Date	**5-03-2022**	