CS 203: OBJECT ORIENTED PROGRAMMING Using C++ (3-0-1)
4 Credits

PART A- THEORY

Faculty: Prof. Shylaja S S **No. of Hours: 52**

Class No.	Chapter Title/ Reference Literature	Topics to be Covered	% of Portions Covered	
			Reference Chapter	Cumulative
1.	**UNIT – I** **CHAP 1** - Page# 6-8, 12-16, **CHAP 2** – Page# 21, 37-39, **CHAP 3** – Page# 45-51, **CHAP 4** – Page# 69-82, **CHAP 5** – Page# 87,88, 94-101, **CHAP 7** – Page# 143-154, 160-162	Introduction	19.23	19.23
2.		**Introduction:** Learning C++, Design of C++,		
3.		Use of C++, C and C++, Programming in C++.		
4.		**A Tour of C++**: What is C++, Object Oriented Programming, Comments and Indentation.		
5.		**A Tour of the Standard Library**: Introduction, Hello World,		
6.		The Standard Library Namespace, Output, Strings, Input.		
7.		**Types and Declarations**: Boolean, Character types, Integer Types, Floating point types, sizes, void, and enumerations, Declarations,		
8.		Pointers, Constants, References and Pointer to Void.		
9.		**Functions in C++:** Function declarations, Argument passing, Value return.		
10.		Overloaded function names, Default arguments, macros**.**		
11.	**UNIT – II** **CHAP 5** - Page# 87-88, 127-130, **CHAP 10** – Page# 223-232, 234-236,	**Dynamic Memory Allocation:** Pointers and the Free store,	19.23	38.46
12.		Allocating and de-allocating memory for arrays, memory exhaustion.		
13.		**Classes :** Introduction, classes – member functions, Access control,		
14.		constructors, static class members,		
15.		copying class objects, constant member functions,		

16.	242-248	self-reference, physical and logical constness, structures and classes,		
17.		In-class function definitions. Objects – Destructors, default constructors,		
18.		construction and destruction, local variables,		
19.		Copying objects - necessary member initialization.		
20.		Programming Examples		
21.	**UNIT - III** **CHAP 11** - Page# 261-272, 275-278, 283-287	**Operator Overloading:** Introduction, Operator functions – Binary and unary operators,	17.31	55.77
22.		Predefined meaning for operators, Operators and user-defined types,		
23.		operators in namespaces, A Complex number type – Member and non-member operators,		
24.		mixed-mode arithmetic, initialization, copying, literals,		
25.		constructors and conversions,		
26.		Conversion operators, Essential Operators,		
27.		Subscripting.		
28.		Programming Examples		
29.	**UNIT - IV** **CHAP 12** - Page# 301-314, **CHAP 15** – Page# 389-399, **CHAP 21** – Page# 605-612, 613-618, 621, 625-629, 637-640, 642-644	**Class Inheritance:** Introduction, derived classes, member functions,	23.08	78.85
30.		Constructors and destructors,		
31.		Copying, Class hierarchies, Type fields,		
32.		Virtual functions, Abstract classes.		
33.		**Class Hierarchies**: Multiple Inheritance – Ambiguity resolution,		
34.		Inheritance and using-declarations, Replicated base classes,		
35.		Overriding, Virtual base classes, programming virtual base classes.		
36.		**Streams:** Introduction**,** Output – output streams, output of built-in types,		
37.		output of user-defined types, Input – input of built-in types, stream state,		
38.		Input of user-defined types,		

		formatting – integer output, floating point output.		
39.		File Streams, closing of streams, Buffering – output streams and buffers, input streams and buffers.		
40.		Programming Examples		
41.	**UNIT - V** **CHAP 11** – Page# 278-282, **Appendix C** – Page# 852-853, **CHAP 13** – Page# 327-338, 341-345, **CHAP 14** – Page# 355-364, 369-378, 380-381	**Friends:** Finding friends**,** Friends and members,	21.15	100
42.		friend classes as members of enclosing classes, friendship in derived classes.		
43.		**Templates:** Introduction, A simple string template –		
44.		Defining a template, template instantiation, and template parameters, type-checking.		
45.		Function templates – Function template arguments, function template overloading,		
46.		Specialization - template function specialization.		
47.		**Exception Handling:** Error Handling, Grouping of exceptions, catching exceptions,		
48.		Resource Management – Exceptions and new,		
49.		Exceptions in constructors and destructors and copy constructors,		
50.		Exception specifications – checking exception specifications,		
51.		Unexpected exceptions, Uncaught exceptions.		
52.		Programming examples.		

LITERATURE:

Book Type	Code	Title & Author	Publication Information	
			Edition	Publisher
Text Book	T1	The C++ Programming Language- Bjarne Stroustrup	3rd	Pearson Education

PART B – LABORATORY

Week No.	Program No
1	Introduction to the lab and sample programs
2	1
3	2
4	3
5	4
6	5
7	6
8	7
9	8
10	9
11	10,11
12	12

Program #1

Create a class Vehicle having non-static data member registration number and a static data member count. Non static member functions setregno() and getregno() are used to get & set the registration number. A static member function getVehiclecount() is used to return the number of vehicles in the garage. Use constructor to increment the vehicle count when a vehicle is created and the destructor to decrement the count when the vehicle is destroyed.

Program #2

Design a class Polar which describes a point in the plane using polar coordinates radius and angle. Overload + and – to add and subtract two Polar objects.

Note: Use the following trigonometric functions to covert polar points into rectangular coordinates. Then add and subtract the rectangular co-ordinates and convert the result into polar co-ordinates.

x= r*cos(a); y=r*sin(a); a=atan(x/y); r=sqrt(x*x*+y*y);

Program #3

Create a class called Numbers consisting of a set of integers and doubles. Use appropriate functions to accept data and print the sum of integers and doubles. Overload the sum() to illustrate function polymorphism

Program #4

Create a class called Inventory having data members name, code and price. Add appropriate member function to read and print items. Use necessary I/O

manipulators for formatting the output. Illustrate how the objects of Inventory class can write to and read from disk files.

Program #5

a. Create a template function to sort a list of elements.

b. Using appropriate ios functions, display the square root of number from 1 to 20 in the following format:

VALUE*********SQRT OF VALUE

+……1……..+1.0000 +……2……..+1.4142

Program #6

Implement a person Class. Each object in this class will represent a Person entity. Data members for person class name, dob, address. Derive another class called employee having its own member basic pay, DA, HRA. Compute net salary. Include constructor, destructor, access functions and print functions

Program #7

Implementation of task to perform basic arithmetic operations on two complex numbers by overloading the arithmetic operators +, -, *, / and I/O stream operators.

Program #8

Implement a class called Person having data members as name, dob and address. Derive a class called from Person called student having data members roll no and semester. Derive another class called Exam from student which has data member's marks1, marks2 and computes the average and displays the topper of the class. Use suitable member functions to accept and display data in these classes.

Program #9

Task to implement an array using class templates. Implement the following operations:

i) comparing two arrays
ii) adding two arrays
iii) finding the max and min in an array
iv) sorting the array
v) display the array object

Program #10

Implement a string class. Each object of this class will represent character string. Data members are length of the string and actual character string. Include constructor, destructor, access functions, print function and subscript function.

Program #11

Task to implement functions for computing harmonic mean and geometric mean. If the means cannot be computed catch these exceptions and suitable handle the same.

Program #12

Task that will show the following menu to the user

i) insert an integer at the end of list
ii) insert an integer at the beginning of the list
iii) insert an integer before a specified integer in the list
iv) delete the first integer from list
v) delete the last integer from the list
vi) delete a specified integer from the list
vii) display the list of integers
viii) save the list of integers
ix) quit

Implement the above menu and associated functions using the list class of the STL.

Question Bank

UNIT I

1	What is Procedure-oriented Programming System?	8
2	What are the main conceptual differences between object-oriented programming and the other programming techniques?	
3	What is Object-oriented Programming System?	8
4	Explain the concept of an object with real life example.	5
5	Differentiate between C and C++	8
6	Briefly explain the mechanics of creating a program.	10
7	Explain the console I/O functions supported by C++.	5
8	What is a function prototype?	5
9	What is the signature of a function	5
10	Explain parameter passing in C++.	6
11	What are inline functions? What are the advantages and disadvantages of inline functions?	6
12	What are the rules behind using default arguments? How does it help the programmer?	6
13	Write the program to tabulate 2 power n ranging n from 1 to 16 using setw manipulator.	8
14	Write a function called reverseit () that reverses a string (array of characters) and passes the string to function from the main as an argument. Write main () to use this function.	8
15	Explain Function polymorphism	5
16	Write a short note on Reference variables and reference as function parameters.	5
17	What is reference variable? Explain it with an example.	6
18	How does a reference variable differ from pointer variable?	6
19	What are pointers and const types in C++?	5
20	What are inline functions? What are its benefits?	5
21	Write a C++ program that asks for a distance in furlongs and converts it to yards (one furlong is 220 yards).	6
22	Write a short note on C++ Statements.	5
23	What is a macro? Compare macros and inline functions.	5
24	Do inline functions improve performance?	5
25	What are inline functions? Explain the different function can be made inline.	5

UNIT II

1	Explain how dynamic memory allocation done in C++?	8
2	Explain new operator?	8
3	Explain delete operator	5

4	Explain the difference between the four objects defined below:
	a) int ival = 1024;;
	b) int *pi = &ival;
	c) int *pi2 = new int(1024);
	d) int *pi3 = new int[1024];
		8

5	How does a class differ from a structure?	10
6	Explain the structure of a C++ class.	5

7	What is an object?	5
8	What is the use of scope resolution operator? Explain with an example.	5
9	Explain the various access specifies available in C++	5
10	Write a brief note on this pointer.	5
11	What is a mutable data member? How it differs from static data member.	5
12	Explain with an example how the C++ class construct provides facilities to implement data abstraction	5
13	What is a constructor? List the different types of constructors available un C++.	5
14	How do we unable a construction function? Explain with an example.	5
15	Write short notes on. a) Copy construction. b) B) Parameterized construction.	10
16	Describe the importance of destructors with an example.	5
17	Can constructors be overloaded? If yes, explain with an example.	5
18	Difference between realloc () and free?	5
19	What is default constructor	5
20	Bring out the difference between class and union.	5
21	In what way is destructor different from delete operator?	5
22	In what way constructor is different from automatic initialization.	5
23	What do you understand by X (X&) constructor	5
24	How is constructor overloading different from function overloading	5
25	Which member functions does the compiler create automatically if the programmer does not include them in the class definition?	5
26	What are parameterized constructors.	5
27	Explain the process of execution of constructors and destructors.	5
28	Explain with an example the use of pointer to a function.	8
29	Write a piece of code that asks user to enter a positive inter value and then create an a dynamic array of that many integers.	8
30	Starting with an array of pointer to strings. Provide a function to sort strings in alphabetical order.	

UNIT III

1	What do you mean by function over riding of derived class?	8
2	Name the operators which are overloaded only as member function of a class	8
3	What are the drawbacks of operator overloading? Justify.	5
4	Write a c++ program to concatenate two strings using the concept of class and operator overloading.	8
5	Explain how memory management operators are used for overloading in C++.	10
6	Assume that an application needs to add and compare string objects + and == operators. Design appropriate class, which does this job.	8
7	Explain how operator [] is overloaded ,with example	5
8	What are some advantages/disadvantages of using friend functions?	5
9	How can we declare a generic pointer pointing to a function?	8
10	Can we overload a destructor? Justify.	5
11	What do you mean by operator overloading and function overloading	6
12	What are the differences between a friend function and member function	6
13	Write a program in c++ to overload the unary operator ++ so that it can be used on objects. Your program should take care of both pre and post increments with any possible assignments	6
14	Define class and object? Explain the components of a class?	8

	Bring out the difference between Structure and a class?	
15	Explain a) Constructor. b) Destructor. c) Inline Function. d) Scope Resolution Operator e) Function Overloading.	10
16	Explain Static Data Members and Static Member Functions with Examples.	8
17	Write a C++ program to create a class called COMPLEX and implement the following by overloading the function ADD which returns the complex number. a) ADD (s1, s2) – where s1 and s2 are complex numbers. b) ADD (s1, s2) – where s1 is an integer (Real Part) and s2 is complex number. Display the result by overloading the operator <<.	10
18	Explain the concept of constructors and destructors. What is default constructor and constructor overload?	10
19	Define a class date with following private member date, day, month and year. Write member functions to initialize date, copy date, increment date, Add or Subtract date and display date. Write a short program that uses this class.	10
20	Create a class called time that has separate in member data for hrs, min and sec. A default constructor to initialize this data to 0, and another should initialize it to fixed values. A member function should add 2 objects to type time and return result. No other member function should display it in he: min: se format. Write main () to create 2 initialize time objects then add and display time.	10
21	Define a class called stack in C++ to simulate the working of a stack. Include push( ), pop( ) and display( ) as class member functions. Overflow and underflow conditions should be taken care in the design.	10
22	Design a class called complex to add subtract, multiply two complex number by overloading +, -, and *. Write these operators overloading functions as friend functions. Add a main program to test this class. Why friend functions is a good choice for this problem.	10
23	Write a C++ program to implement bank account member function which should allow the following Assigning starting values to the data member. Assigning starting values to the data member Printing an amount of money given by an argument. Withdraw an amount of money given by an argument	10
24	Write an interactive program in c++ to simulate the working of a 4function calculator (+, -, *, /). The program should request the user to enter a number followed by an operator and again a number. It should then carry out the specified operation and display the result when the user presses =.	10
25	Explain static data members and static member functions.	5
26	Explain Nested classes with example.	5
27	If ObjA belongs to class A and ObjB belongs to class B. Write a c++ program to enable the user to write statement like ObjA=ObjB	10
28	What is operator overloading? Why do we need it?	5
29	Explain how to overload unary and binary operators, in case of unary operator how would you take care of pre and post operations	10
30	Write a C++ program to create a class STRING and implement the following. Display the results by overloading the operator << 1. STRING s1 = "PES" 2. STRING s2 = "IT" 3. STRING s3 = s1+s2;	10

	4. STRING s4 = s3+"College" STRING s5 = "College"+s4, using copy constructor	
31	Write a C++ program to create a class called STACK using array of integers. Implement the following operations by overloading the operators "+" and "-". Also display the status and contents of the stack after each operation by overloading the operator <<. 1. s1=s1+element; where s1 is an object of the class STACK and element is an integer to be pushed on top of the stack. 2. s1 = s1--; where s1 is an object of the class STACK, -- operator pops the element. Handle STACK empty and STACK full conditions.	10

32	Write a C++ program to create a class called MATRIX using two dimensional array of integers. Implement the following by overloading the operators == which checks the compatibility of two matrices to be added and subtracted. Perform the following by overloading + and – operators. Display the result by overloading the operator <<.
	If (m1 == m2)
	{
	m3 = m1+m2;
	m4 = m1-m2;
	}
	else
	"display error",
	where m1,m2,m3,m4 are MATRIX objects
		10

33	What are the advantages of inheritance	5
34	What is inheritance?	5
35	In what ways does inheritance affects the size and behavior of derived class abject.	5

UNIT IV

1	How can the members of base class be accessed in derived class?	8
2	Can a derived class pointer point to a base class?	8
3	How can a derived class pointer forcibly made to point at an object of derived class?	5
4		8
5	What are different of inheritance? That is available in C++?	10
6	What is multiple inheritances? What kind of ambiguities does a multiple inheritances lead to? How can they be removed?	8
7	Write short notes on. a) Multi-level inheritance b) Multiple inheritance c) Hybrid level inheritance	8
8	In which order are the constructors and destructors called when an object of derived class is created?	5
9	What is a virtual function?	6
10	Explain how dynamic (runtime) poly morphisn is achieved using virtual functions.	8
11	When do we make a virtual function 'Pure'?	8
12	What is a pure virtual function? Explain with an example.	5
13	How does the compiler resolve a call to a virtual function?	6
14	What is a stream? Explain with a neat diagram the C++ class hierarchy for stream handling.	6

15	Explain text made & binary made LIP w.r.t to i) Character data ii) Numeric data	6
16	Different between text files & binary file.	5
17	Explain the various functions available for text ILO in C++.	5
18	Explain the read () and write () function.	5
19	Explain the Open () function along with the various modes supported by it.	6
20	What is a fule pointer? How can the file pointers can be explicitly manipulators.	6
21	Write a short notes on i) SeeKP() ii) tellP() iii) Seekg() iv) tellp()	10
22	How can a file be opened for both reading & writings?	5
23	What is the difference between opening a file using the constructor of stream class and open () function.	5
24	Explain with an example, how a file be randomly accessed C++.	8
25	Write short note on i)eof() ii) fail() iii) bad() iv)Clear().	6
26	Explain with suitable examples the pre-defined manipulators available in C++	5

UNIT V

1	Explain the syntax for overloading operators using. i) Member function ii) Friend function	8
2	How are arithmetic operators overloading using? i) Member function ii) Friend function	8
3	What is a function template? Explain with its syntax.	5
4	WAP to swap two numbers using template.	8
5	What is a class template? Explain with an example.	10
6	Write a brief on STL.	5
7	What a list class? Explain the various functions available this class.	6
8	What are function templates? How do they work?	6
9	What is Function Template? Explain purpose of function template with suitable example?	6
10	Why do we need function template explicit specialization, explain with an example?	8
11	Write a C++ program to create a template function for Quicksort and demonstrate the sorting of integers and double data types	8
12	What are some advantages/disadvantages of using friend functions?	5
13	How can we declare a generic pointer pointing to a function?	6
14	How do you write a function, which can take template arguments?	6
15	What is exception handling?	5
16	Write a short note on a) constructor exceptions b) Exception Specifications c) The default handler d) Throwing objects	10