



Chapter 1 : Object Oriented Methodology	1-1 to 1-94
✓ Syllabus Topic : Introduction.....	1-1
1.1 Introduction	1-1
✓ Syllabus Topic : Advantages and Disadvantages of Procedure Oriented Language	1-2
1.2 Advantages and Disadvantages of Procedure Oriented Language (April 2017).....	1-2
✓ Syllabus Topic : What is object oriented ? What is object Oriented Development ?.....	1-3
1.3 Object Oriented Programming its Development (April 2017).....	1-3
✓ Syllabus Topic : Benefits and Applications of OOPS	1-4
1.4 Benefits and Applications of OOPS.....	1-4
1.4.1 Benefits and Applications of OOPs (April 2017)	1-5
✓ Syllabus Topic : Principles of OOPS : OOPS Paradigm	1-5
1.5 Principles of OOPS	1-5
1.5.1 Principles of Object Oriented Programming (OOPS)	1-5
✓ Syllabus Topic : Basic concepts of OOPS.....	1-6
1.5.2 Basic concepts of OOPS. (April 2017)	1-6
1.6 Overview of Tokens of C / C++.....	1-7
1.6.1 Character Set of C / C++	1-7
1.6.2 Keywords	1-9
1.6.3 Identifiers	1-10
1.6.4 Data Types	1-11
1.6.5 Constants and Variables.....	1-13
1.6.6 Escape Sequences	1-13
1.6.7 Operators.....	1-14
1.7 Program Structure	1-14
1.7.1 Dynamic Initialization of Variables.....	1-16
1.7.2 enum.....	1-16
1.7.3 Namespace	1-18
1.8 Integrated Development Environment (IDE).....	1-19
1.9 Operators	1-20
1.9.1 Unary Operators.....	1-20
1.9.2 Binary Operators.....	1-23
1.9.3 Ternary Operator.....	1-27
1.9.4 Assignment Operators.....	1-28
1.9.5 Selection Operators.....	1-29
1.9.6 new operator	1-30
1.10 Precedence and Associativity of Operators.....	1-30
1.11 Solved Examples	1-32
1.12 Formatted and Unformatted IO Functions	1-35
1.12.1 Formatted IO Functions	1-35
1.12.2 Unformatted IO Functions	1-37



1.13	C++ IO Techniques	1-37
1.14	Manipulators.....	1-38
1.15	Control Structure : for Loop.....	1-40
1.15.1	Programs Based on for Loop	1-41
1.15.2	Nested for Loop	1-42
1.16	Control Structure : while and do-while Loops.....	1-43
1.17	Control Structure : if-else Selective Statement	1-47
1.17.1	Programs using if-else Statement	1-48
1.18	Control structure : Switch-Case Selective Statement.....	1-50
1.19	Branching Statements (break, continue and goto)	1-53
1.20	Simple Functions.....	1-55
1.20.1	Default Arguments in a Function.....	1-59
1.21	Recursive Functions	1-60
1.22	Arrays	1-64
1.23	Multi-dimensional Arrays	1-67
1.24	Strings.....	1-70
1.25	Pointers.....	1-72
1.25.1	Referencing and De-referencing (Operators in Pointers)	1-73
1.26	Programs Related to Pointers	1-74
1.26.1	Simple Referencing and De-referencing using Pointers.....	1-74
1.26.2	Pointer to Pointer	1-75
1.26.3	Increment and Decrement Operators Operation on Pointer Variables	1-77
1.26.4	Pointer to Array	1-78
1.26.5	Call and Return by Value, Address and Reference	1-79
1.27	Structures.....	1-85
1.28	Arrays of Structure Variable	1-88
1.29	Nested Structures.....	1-91
1.30	Unions	1-92
1.31	Comprehensive University Questions and Answers	1-94

Chapter 2 : Classes and Objects**2-1 to 2-32**

✓ Syllabus Topic : Simple Classes (Class specification, Classmembers accessing)	2-1
2.1 Simple Classes (Class specification, Classmembers accessing) (April 2017)	2-1
✓ Syllabus Topic : Defining Member Functions.....	2-1
2.2 Defining Member Functions	2-1
2.2.1 Internally Defined Functions	2-2
2.2.2 Externally Defined Functions	2-6
2.2.3 Inline Functions (April 2017).....	2-10
✓ Syllabus Topic : Returning object from functions, friend classes, Pointer to object, Array of object	2-14
2.3 Friend Classes, passing object as an argument and returning object from a function (April 2017)	2-14



2.3.1	Array of pointer to object.....	2-20
✓	Syllabus Topic : Introduction, Default constructor, parameterized constructor.....	2-22
2.4	Introduction to Constructors (April 2017).....	2-22
2.4.1	Default Constructor.....	2-23
2.4.2	Parameterized Constructor (April 2017).....	2-26
2.5	Destructors (April 2017).....	2-29
2.6	Comprehensive University Questions and Answers.....	2-31
Chapter 3 : Polymorphism		3-1 to 3-40
✓	Syllabus Topic : Concept of Function Overloading	3-1
3.1	Concept of Function Overloading (April 2017).....	3-1
✓	Syllabus Topic : Overloaded Operators or Data Conversion between Objects and Basic Types ...	3-7
3.2	Overloaded Operators or Data Conversion between Objects and Basic Types (April 2017)	3-7
✓	Syllabus Topic : Overloading Unary, Binary, Comparison and arithmetic assignment Operators .	3-8
3.3	Overloading Unary, Binary, Comparison and arithmetic assignment Operators (April 2017)..	3-8
✓	Syllabus Topic : Virtual Functions Introduction and need	3-23
3.4	Virtual Functions.....	3-23
✓	Syllabus Topic : Virtual and Pure Virtual Function	3-24
3.4.1	Virtual and Pure Virtual Function (April 2017).....	3-24
3.4.2	Implementation of dynamic binding using Pointer to Object and Array of pointer to object (April 2017).....	3-27
✓	Syllabus Topic : Abstract classes.....	3-32
3.4.3	Virtual Base Class and Abstract Class (April 2017).....	3-32
3.4.4	Virtual Destructors (April 2017).....	3-35
✓	Syllabus Topic : Static Function	3-35
3.5	Static Function.....	3-35
✓	Syllabus Topic : This Pointer.....	3-37
3.6	This Pointer	3-37
3.7	Comprehensive University Questions and Answers.....	3-39
Chapter 4 : Program Development using Inheritance		4-1 to 4-37
✓	Syllabus Topic : Introduction, Understanding Inheritance	4-1
4.1	Introduction and Understanding Inheritance or containership and its advantages (April 2017).....	4-1
✓	Syllabus Topic : Choosing the access specifier	4-2
4.1.1	Private, Protected and public Members (April 2017)	4-2
4.1.2	Single Inheritance	4-3
✓	Syllabus Topic : Multi Level Inheritance	4-8
4.1.3	Multi Level Inheritance (April 2017).....	4-8
✓	Syllabus Topic : Multiple Inheritance.....	4-14



4.1.4	Multiple Inheritance (April 2017).....	4-14
✓ Syllabus Topic :	Hybrid Inheritance	4-18
4.1.5	Hybrid Inheritance (April 2017)	4-18
4.1.6	Problem in Multiple and Hybrid Inheritance.....	4-22
4.1.7	Hierarchical Inheritance.....	4-25
4.1.8	Need of Constructors in a Derived Class.....	4-30
✓ Syllabus Topic :	Exception Handling : Introduction	4-31
4.2	Exception Handling.....	4-31
✓ Syllabus Topic :	Exception Handling Mechanism.....	4-31
4.3	Exception Handling Mechanism	4-31
4.3.1	Try-catch-throw, Multiple Catch and Catch All (April 2017)	4-31
4.3.2	Implementing User Defined Exception	4-34
✓ Syllabus Topic :	Concept of Throw and Catch with Example.....	4-35
4.4	Concept of Throw and Catch with Example	4-35
4.4.1	Rethrowing Exception (April 2017).....	4-35
4.5	Comprehensive University Questions and Answers	4-37
Chapter 5 : Templates		5-1 to 5-31
✓ Syllabus Topic :	Templates : Introduction to Templates	5-1
5.1	Introduction to Templates (April 2017)	5-1
✓ Syllabus Topic :	Function Templates and Examples	5-1
5.2	Function Templates and Examples (April 2017).....	5-1
5.2.1	Overloading a Template Function Using a Non-template Function.....	5-12
5.2.2	Overloading a Template Function Using a Template Function.....	5-13
✓ Syllabus Topic :	Class Template and Examples	5-14
5.2.3	Generic Classes, Class Template and Examples	5-14
5.2.4	Overview and use of Standard Template Library (STL)	5-15
✓ Syllabus Topic :	Working with files : Introduction	5-15
5.3	Working with Files (April 2017)	5-15
5.3.1	Advantages of Stream Classes.....	5-17
✓ Syllabus Topic :	File Pointer and their Manipulation.....	5-17
5.4	File operations, File Pointer and their Manipulation (April 2017).....	5-17
5.5	Random and Binary File Handling	5-21
5.6	Exceptions and Error Handling During File Operations.....	5-29
5.7	Comprehensive University Questions and Answers	5-30
	• Model Question Papers :	M-1 to M-4
	• List of Practicals :	P-1 to P-58
	• Appendix A : Solved University Question Papers of Nov. 2017 and April 2018 ..	A-1 to A-11
	• Appendix B : Solved University Question Papers of Nov. 2018 and April 2019 ..	B-1 to A-13

**LIST OF PROGRAMS**

Program No.	Name of the Program	Page Nos.
Program 1.6.1	A list of valid and invalid identifiers is given below with reasons wherever required. 1. simple_interest : Valid 2. char : Invalid, because it is a keyword 3. 3friends: Invalid, because starts with a digit 4. _3friends : Valid 5. Simple interest: Invalid, because blank spaces are not allowed 6. #3friends : Invalid, because no special symbol except underscore is allowed. 7. void : Invalid, because keyword not allowed. 8. Void : Valid, case sensitive.	1-10
Program 1.7.1	Write a simple program to display a statement "Hello Friend".	1-14
Program 1.14.1	Write a C / C++ program to accept a number and display its square.	1-39
Program 1.15.1	Write a program to display the word "Computer" five times using for loop.	1-41
Program 1.15.2	Write a program to display the following : * ** *** **** *****	1-42
Program 1.17.1	Write a program to check if the entered number is prime number or not.	1-48
Program 1.18.1	Write a program to display the user entered single digit number in words.	1-51
Program 1.20.1	Write a program to add two numbers using function.	1-56
Program 1.20.2	Write a program to add two numbers using a void function.	1-60
Program 1.21.1	Write a program to find the factorial of a number, using a recursive function.	1-61
Program 1.22.1	Write a program to accept 'n' integers from user into an array and display them one in each line.	1-65
Program 1.23.1	Write a program to accept an $m \times n$ matrix and display it in natural form.	1-68
Program 1.24.1	Write a program to accept and display a string.	1-71
Program 1.26.1	Write the output of the following program.	1-74
Program 1.26.2	Write the output of the following program.	1-75
Program 1.26.3	Write the output of the following program.	1-77
Program 1.26.4	Write the output of the following program.	1-78
Program 1.26.5	Write a program to swap two numbers using a function. Pass the values to be swapped to this function using call-by-value method.	1-80

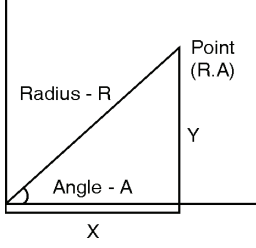


Program No.	Name of the Program	Page Nos.
Program 1.26.6	Write a program using function with arguments to swap the values of a pair of integers.	1-82
Program 1.26.7	Write a program using a function with arguments to swap the values of a pair of integers using call by reference.	1-83
Program 1.27.1	Write a program to store and display the name, roll number and fees of a student using structure.	1-86
Program 1.28.1	Write a program to store the name, roll number and marks in three subjects of 'n' students using structure. Generate a merit list with respect to the total marks scored i.e. display the output in tabular form in order of maximum total marks to minimum total marks in three subjects.	1-88
Program 2.2.1	Write a program to find area of circle using Object Oriented Programming such that the class circle must have three member functions namely : (a) read() to accept the radius from the user. (b) compute() for calculating the area. (c) display() for displaying the result.	2-2
Program 2.2.2	Write a program to calculate the value of the following series using internal member function : $S = 1^2 + 2^2 + 3^2 + 4^2 \dots + n^2$	2-5
Program 2.2.3	Write a program to find area of circle using Object Oriented Programming such that the class circle must have three externally defined member functions namely : (a) read() to accept the radius from the user. (b) compute() for calculating the area. (c) display() for displaying the result.	2-7
Program 2.2.4	Write a program to calculate the value of the following series using external member function : $S = 1^2 + 2^2 + 3^2 + 4^2 \dots + n^2$	2-9
Program 2.2.5	Write a program to find area of circle using Object Oriented Programming such that the class circle must have three inline functions namely : (a) read() to accept the radius from the user (b) compute() for calculating the area. (c) display() for displaying the result.	2-11
Program 2.2.6	Write a program to calculate the value of the following series using inline member function : $S = 1^2 + 2^2 + 3^2 + 4^2 \dots + n^2$	2-12
Program 2.3.1	Write a program to add two complex numbers using operator overloaded by a friend function.	2-15
Program 2.3.2	Write a program to add two complex numbers using a friend function to add the complex numbers.	2-17



Program No.	Name of the Program	Page Nos.
Program 2.3.3	Define a class numbers, having data members as A and B. Define a friend function mul () to multiply these two numbers and display all numbers using display () friend function.	2-18
Program 2.3.4	Write a program to find area of circle using Object Oriented Programming such that the class circle must have three inline functions namely : (a) read() to accept the radius from the user. (b) compute() for calculating the area. (c) display() for displaying the result. Make an array of pointers to object	2-20
Program 2.4.1	Write a program to find area of circle using Object Oriented Programming. The value of the radius must be accepted from the user in the constructor and the class circle must have two inline functions namely : (a) compute() for calculating the area. (b) display() for displaying the result.	2-23
Program 2.4.2	Write a program to calculate the value of the following series using default constructor and inline member function :	2-25
Program 2.4.3	Write a program to find area of circle using Object Oriented Programming. The value of the radius must be accepted from the user in the main program and passed to the parameterized constructor and the class circle must have two inline functions namely : (a) compute() for calculating the area. (b) display() for displaying the result.	2-27
Program 2.5.1	Write a program to demonstrate the destructor.	2-29
Program 3.1.1	Write a program to add two numbers using function overloading such that one function adds two integers, second function adds two float numbers and the third function adds a float number with an integer.	3-1
Program 3.1.2	Write a program to calculate the area of triangle, rectangle and circle using function overloading. The program should be menu-driven.	3-3
Program 3.1.3	Write a program to overload function 'concat' which will concatenate: (i) two strings (ii) string and int	3-5
Program 3.3.1	Write a program to negate (unary operator overloading) the values of two variables contained in an object.	3-8
Program 3.3.2	Write a C++ Program to overload unary minus operator using member function and friend function.	3-10
Program 3.3.3	Write a program to overload unary operators + +(increment) and - - (decrement).	3-11



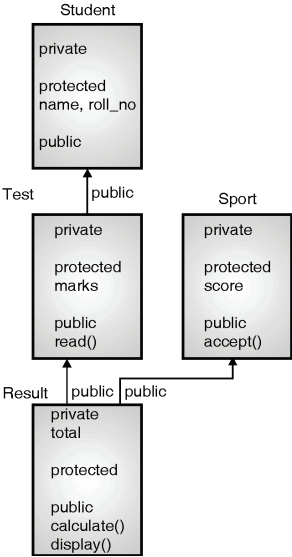
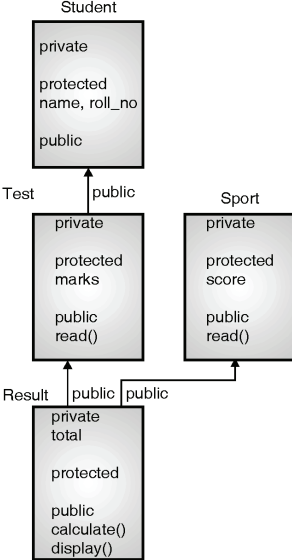
Program No.	Name of the Program	Page Nos.
Program 3.3.4	Write a program to overload binary operator “+” to add two complex numbers.	3-13
Program 3.3.5	Write a program to overload binary assignment operator “+=” to add two complex numbers.	3-15
Program 3.3.6	Write a program to add two distances entered by the user in feet and inches using overload binary operator “+”.	3-16
Program 3.3.7	Define a circle class with radius as data member, necessary constructors and member function to compute area of circle. Class should overload the comparison operator = = to compare two circle objects whether they are equal in radius. Demonstrate its use in main().	3-18
Program 3.3.8	Write a class to create array objects with given size. Overload the binary operator * to multiply the elements of an array object with scalar value. Use them in main() to perform operation like s*a1, where s is scalar value and a1 is array object.	3-20
Program 3.3.9	<p>Design a class Polar which describes a point in the plane using Polar coordinates radius and angle. A point in Polar coordinates is shown in Fig. P. 3.3.9.</p>  <p style="text-align: center;">Fig. 3.3.9</p> <p>Use the overloaded + operator to add two objects of Polar. You need to use the following trigonometric formula. $X = R * \cos(A)$ $Y = R * \sin(A)$ $A = \text{atan}(Y/X) // \text{arc tangent}$ $R = \sqrt{X^2 + Y^2}$</p>	3-21
Program 3.4.1	Write a program to demonstrate function overriding.	3-25
Program 3.4.2	Write a program to demonstrate dynamic binding using virtual function.	3-27
Program 3.4.3	Write a program to demonstrate pure virtual function.	3-30
Program 3.4.4	Write a program to demonstrate virtual class or abstract class.	3-33
Program 3.5.1	Write a program to find how many objects of a class has been created using static member function.	3-35



Program No.	Name of the Program	Page Nos.
Program 3.6.1	Write a program to demonstrate the use of "this" pointer.	3-37
Program 4.1.1	Write a program to add two numbers using single inheritance such that the base class function must accept the two numbers from the user and the derived class function must add these numbers and display the sum.	4-3
Program 4.1.2	Write a program to find the area of circle using single inheritance such that the base class function must accept the radius from the user and the derived class function must calculate and display the area.	4-6
Program 4.1.3	Write a program to calculate percentage of a student using multi level inheritance. The base class function will accept the marks in three subjects from user. A class will be derived from the above mentioned class that will have a function to find the total marks obtained and another class derived from this will have functions to calculate and display the percentage scored.	4-8
Program 4.1.4	Write a program to calculate volume of sphere using multi level inheritance. The base class function will accept the radius from user. A class will be derived from the above mentioned class that will have a function to find the area of a circle and another class derived from this will have functions to calculate and display the volume of the sphere.	4-11
Program : 4.1.5	<p>Write a program to define the following relationship using multiple inheritance.</p> <pre>classDiagram class Polygon { protected height, width; public void read(int a, int b); } class Rectangle { public int area(); } class Triangle { public int area(); } class Output { public void output(int); } Polygon < -- Rectangle Polygon < -- Triangle Polygon < -- Output</pre>	4-15

Fig. P. 4.1.5 : Class Diagram of Multiple Inheritance for Program 4.1.5



Program No.	Name of the Program	Page Nos.
Program 4.1.6	<p data-bbox="469 434 1257 495">Write a program to define the following relationship using hybrid inheritance.</p>  <pre data-bbox="715 501 1007 1055">classDiagram class Student { private protected name, roll_no public } class Test { private protected marks public read() } class Sport { private protected score public accept() } class Result { private total protected public calculate() display() } Student < -- Test Student < -- Sport Test < -- Result Sport < -- Result</pre> <p data-bbox="549 1070 1177 1131">Fig. P.4.1.6 : Class Diagram of Hybrid Inheritance for Program 4.1.6</p>	4-19
Program 4.1.7	<p data-bbox="469 1169 1257 1229">Write a program to define the following relationship using hybrid inheritance.</p>  <pre data-bbox="715 1236 1007 1794">classDiagram class Student { private protected name, roll_no public } class Test { private protected marks public read() } class Sport { private protected score public read() } class Result { private total protected public calculate() display() } Student < -- Test Student < -- Sport Test < -- Result Sport < -- Result</pre> <p data-bbox="549 1809 1177 1870">Fig. P.4.1.7 : Class Diagram of Hybrid Inheritance for Program 4.1.7</p>	4-22



Program No.	Name of the Program	Page Nos.
Program 4.1.8	<p>Write a program to define the following inheritance relationship.</p> <pre>classDiagram class Staff { code name } class Teacher { subject experience } class Typist { speed experience } class Officer { grade department } class Regular { salary } class Casual { daily wages } Staff < -- Teacher Staff < -- Typist Staff < -- Officer Typist < -- Regular Typist < -- Casual</pre> <p>Fig. P.4.1.8 : Class Diagram of Hierarchical Inheritance for Program 4.1.8</p>	4-26
Program 4.3.1	Write a program to demonstrate the use of try catch block with the arguments as an integer and a string using multiple catch blocks.	4-33
Program 4.3.2	Write a program to accept password and throw an exception if the password has less than 6 characters or does not contain a digit.	4-34
Program 4.4.1	Write a program to accept password and throw an exception if the password has less than 6 characters or does not contain a digit. Give another chance to enter a correct password and rethrow an exception if the password is again incorrect.	4-35
Program 5.2.1	Write a program to write a generic function or template and demonstrate addition of multiple types of data using the same.	5-2
Program 5.2.2	Write a program to write a generic function or template and demonstrate swapping of multiple types of data using the same	5-3
Program 5.2.3	Write a C++ program using a class template to read any five parameterized data type such as float and integer and print the average.	5-4
Program 5.2.4	Write a program to create a vector class template to add, delete and display values from the vector.	5-5
Program 5.2.5	Write a C++ program using a class template to read any parameterized data type such as float and integer and print them in sorted form.	5-9
Program 5.2.6	Write a program to write a generic function or template and demonstrate addition of multiple types of data using the same. Write overloaded function for showing addition of char type data.	5-12



Program No.	Name of the Program	Page Nos.
Program 5.2.7	Write a program to write a generic function or template and demonstrate addition of multiple types of data using the same. Write overloaded template for showing addition of char type data and integer type data.	5-13
Program 5.4.1	Write a program to write and read a string from/to a file.	5-19
Program 5.4.2	Write a program to write and read string, integer and float from/to a file	5-20
Program 5.5.1	Write a program to write string, integer and float to a file and read it as binary data	5-21
Program 5.5.2	Write a program to store employee names with their designation and netpay to a file on console. Also display employee details from file. Create a class 'employee'	5-22
Program 5.5.3	Write a program to display the contents of a text file in the reverse order (use pointer manipulation)	5-24
Program 5.5.4	Write a C++ program to read character data from a file. Create one file to store all capital alphabets and another file to store all small case alphabets. Also display contents of both files.	5-25
Program 5.5.5	Write a c++ program which opens two text files (only if they exist) and concatenates the contents of the second file to the first. Display an error message if the file do not exist	5-26
Program 5.5.6	A file "student.txt" contains roll numbers and names. Write a C++ program to read the contents of this file and search for a student having a specific roll number.	5-27

List of Practicals

Name of the Program	Name of Program	Page Nos.
1.	Classes and methods	P-1
Program 1(a)	Design an employee class for reading and displaying the employee information, the getInfo() and displayInfo() methods will be used repectively. Where getInfo() will be private method	P-1
Program 1(b)	Design the class student containing getData() and displayData() as two of its methods which will be used for reading and displaying the student information respectively. Where getData() will be private method.	P-2
Program 1(c)	Design the class Demo which will contain the following methods: readNo(), factorial() for calculating the factorial of a number, reverseNo() will reverse the given number, is Palindrome() will check the given number is palindrome, is Armstrong() which will calculate the given number is armStrong or not. Where readNo() will be private method	P-3



Name of the Program	Name of Program	Page Nos.
Program 1(d)	Write a program to demonstrate function definition outside class and accessing class members in function definition	P-5
2. Using friend functions		P-6
Program 2(a)	Write a friend function for adding the two complex numbers, using a single class	P-6
Program 2(b)	Write a friend function for adding the two different distances and display its sum, using two classes	P-8
Program 2(c)	Write a friend function for adding the two matrix from two different classes and display its sum.	P-10
3. Constructors and method overloading		P-13
Program 3(a).	Design a class Complex for adding the two complex numbers and also show the use of constructor.	P-13
Program 3(b)	Design a class Geometry containing the methods area() and volume() and also overload the area() function	P-15
Program 3(c)	Design a class StaticDemo to show the implementation of static variable and static function	P-17
4. Operator Overloading		P-18
Program 4(a)	Overload the operator unary(-) for demonstrating operator overloading	P-18
Program 4(b)	Overload the operator + for adding the timings of two clocks, And also pass objects as an argument.	P-20
Program 4(c)	Overload the + for concatenating the two strings. For e.g “Py” + “thon” = Python	P-21
5. Inheritance		P-23
Program 5(a)	Design a class for single level inheritance using public and private type derivation.	P-23
Program 5(b)	Design a class for multiple inheritance.	P-25
Program 5(c)	Implement the hierarchical inheritance.	P-28
6. Virtual functions and abstract classes		P-33
Program 6(a)	Implement the concept of method overriding	P-33
Program 6(b)	Show the use of virtual function	P-35
Program 6(c)	Show the implementation of abstract class	P-37



Name of the Program	Name of Program	Page Nos.
7. String handling		P-39
Program 7(a)	String operations for string length , string concatenation	P-39
Program 7(b)	String operations for string reverse, string comparison	P-41
Program 7(c)	Console formatting functions.	P-43
8. Exception handling		P-47
Program 8(a)	Show the implementation of exception handling	P-47
Program 8(b)	Show the implementation for exception handling for strings	P-48
Program 8(c)	Show the implementation of exception handling for using the pointers.	P-49
9. File handling		P-50
Program 9(a).	Design a class FileDemo open a file in read mode and display the total number of words and lines in the file.	P-50
Program 9(b)	Design a class to handle multiple files and file operations	P-51
Program 9(c)	Design a editor for appending and editing the files	P-53
10. Templates		P-54
Program 10(a)	Show the implementation for the following	P-54
Program 10(b)	Show the implementation of template class library for swap function.	P-55
Program 10(c)	Design the template class library for sorting ascending to descending and viceversa	P-56