

| | | | | | | | |
|-----------------------------|--|----------|----------|----------|----------|----------|----------------|
| 20CSP152 | Object Oriented Programming Using C++ Lab | L | T | P | S | C | C H |
| Version 1.00 | | 0 | 0 | 4 | 0 | 2 | |
| Pre-requisites/ Exposure | Basic knowledge of Computer Programming | | | | | | |
| Co-requisites | | | | | | | |

Unit I: Fundamentals of C++

10 Contact Hours

Practical 1.1: WAP to find average marks of five subjects of a student in a class.

Practical 1.2: WAP to swap first and last digits of any number. (For ex:-n=12345, Output:- 52341).

Practical 1.3: WAP to generate the Fibonacci series up to user specified limit. Write all the missing terms (e.g. 4, 6, 7, 9, 10, 11, 12, 14, 15...) also at the end.

Practical 2.1: WAP to input a matrix of dimension m*n. If base address is 1000. Find the address of (m-1, n-1) element of the matrix.

Practical 2.2: Create a class called employee with the following details as variables within it.

1. Name of the employee (string)
2. Age (int)
3. Designation (string)
4. Salary (double)

Write a program to create array of objects for the same to access these. Also, make use of member functions to accept values and print the name, age, designation and salary.

Practical 2.3:WAP to illustrate the use of scope resolution operator. Display the various values of the same variables declared at different scope levels.

Practical 3.1: Write a program to find the largest& smallest of three numbers. (Use inline function MAX and MIN).

Practical 3.2:A dining hall can accommodate only 50 guests. Create a class to store seat number (Generated Automatically) and name of the guests who are seated on first come first seated basis. Define functions to display name of all guests along with seat number. Write a program to show the working of this class using the concept of static data member and static function.

Practical 3.3: WAP to swap private data members of classes named as class_1, class_2 using friend function.

Practical 3.4: WAP to create a class complex to represent complex numbers. The complex class should use a function to add two complex numbers which are passed as arguments. The function should return an object of type complex representing the sum of two complex

numbers.

Practical 4.1: WAP to find area of rectangle using constructor overloading. Also define destructor to delete the memory allocated to objects.

Practical 4.2: WAP to create database of the following items: Name of the student (String), Roll number of the student (int), Height of the student (cm), Weight of the student (kg/gms)

- 1) Create a Constructor to initialize values
 - 2) Create display () function to display the details
 - 3) Illustrate the use of copy constructor
 - 4) Also implement the concept of destructor.
-

Unit-II: Inheritance, Polymorphism, Pointers & Virtual Functions 10 Contact Hours

Practical 5.1: Write a program that takes information about institute staff information for

- 1) Teacher code, name, subject and publication
- 2) Officer code, name and grade
- 3) Typist code, name, speed and daily wages and displays it using multiple inheritance.

Practical 5.2: Create a class student having student uid and getnumber(),putnumber() as member functions to get the values and display it. Derive a class test having marks in different subjects and getmarks() and putmarks() as member functions to get and display the values. Derive another class sports from student class having sports score and getscore(), putscore() as member functions to get and display the values. Derive a class result from test and sports class and define a function display() to calculate total marks. Implement it with the object of result class. If it gives any error, resolve it by adding the required functionality.

Practical 5.3: WAP to illustrate how the constructors are implemented and the order in which they are called when the classes are inherited. Use three classes named alpha, beta, gamma such that alpha, beta are base class and gamma is derived class inheriting alpha & beta. Pass four variable to gamma class object which will further send one integer variable to alpha(),one float type variable to beta().Show the order of execution by invoking constructor of derived class.

Practical 6.1: WAP to calculate and display cube of an integer and float variable using function overloading.

Practical 6.2: Program to demonstrate the unary operator overloading for operator ++. Make a class test. Create a default constructor to initialize the variable.

- 1) Overload operator ++ (Pre) with definition to pre-decrement the value of a variable
- 2) Overload operator ++ (post) with definition to post-decrement the value of variable.

Practical 6.3: WAP for creating a matrix class which can handle integer matrices of different dimensions. Overload the operator (+) for addition and (==) comparison of matrices.

Practical 6.4: WAP to create a class Pairs. Objects of type Pairs can be used in any situation where ordered pairs are needed. Our Task is to overload operator >> and << so that objects of class Pairs are to be input and output in the form (5,3) (5,-6) (-5,6) or (-5,-3). There is no need to implement any constructor/method .

Practical 7.1: WAP to create a class that will maintain the records of person with details (Name and Age) and find the eldest among them. The program must use this pointer to return the result by overloading > operator among two objects.

Practical 7.2: WAP to access members as mentioned in practical 2.2 using pointer to object members.

Practical 7.3: WAP to design a class representing the information regarding digital library (books, tape: book & tape should be separate classes having the base class as media). The class should have the functionality for adding new item, issuing, deposit etc. The program should link the objects with concerned function by the concept of runtime polymorphism.

Unit-III: Exception Handling, DMA & Files 10 Contact Hours

Practical 8.1: WAP to perform exception handling for Divide by zero Exception.

Practical 8.2: WAP to implement the exception handling with the functionality of testing the throw restrictions.

Practical 9.1: WAP to calculate sum of marks of n students of a class inputted via dynamic memory allocation.

Practical 9.2: WAP to allocate memory dynamically for an object of a given class using class's constructor.

Practical 10.1: WAP to copy the contents of one file to another and display it on output screen.

Practical 10.2: WAP to read the class object of student info such as name, age and rollno from the keyboard and to store them on a specified file using read() and write() functions. Again the same file is opened for reading and displaying the contents of the file on the screen.

Suggestive Readings:

1. [C++ Tutorial in PDF for Beginners \(cppforschool.com\)](http://cppforschool.com)
2. [cpp_tutorial.pdf \(tutorialspoint.com\)](http://tutorialspoint.com/cpp_tutorial.pdf)
3. [C++ Notes: Table of Contents \(fsu.edu\)](http://fsu.edu)
4. [basicsCC++1.jnt \(utdallas.edu\)](http://utdallas.edu)

Books and References:

1. E Balagurusamy., "Object Oriented Programming in C++", Tata McGraw-Hill.
2. Robert Lafore, "Object Oriented Programming in C++", Waite Group.
3. Herbert Schildt , "C++- The Complete Reference", Tata McGraw-Hill 2003, New Delhi.
4. Bjarne Stroustrup: "The C++ Programming Language" (4th Edition). Addison-Wesley.
5. Ravichandran , "Programming with C++",Tata McGraw-Hill Education.
6. Joyce M. Farrell," Object Oriented Programming Using C++", Learning.
7. Programming Languages: Design and Implementation (4th Edition), by Terrence W. Pratt, Marvin V. Zelkowitz, Pearson.
8. Programming Language Pragmatics, Third Edition, by Michael L. Scott, Morgan Kaufmann.