

<b>B. Sc. (Information Technology)</b>		<b>Semester – II</b>	
<b>Course Name: Object Oriented Programming Practical</b>		<b>Course Code: USIT2P1</b>	
<b>Periods per week (1 Period is 50 minutes)</b>		<b>3</b>	
<b>Credits</b>		<b>2</b>	
		<b>Hours</b>	<b>Marks</b>
<b>Evaluation System</b>	<b>Practical Examination</b>	<b>2½</b>	<b>50</b>
	<b>Internal</b>	<b>--</b>	<b>--</b>

<b>List of Practical: To be implemented using object oriented language</b>	
<b>1.</b>	<b>Classes and methods</b>
a.	Design an employee class for reading and displaying the employee information, the getInfo() and displayInfo() methods will be used respectively. Where getInfo() will be private method
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.
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, isPalindrome() will check the given number is palindrome, isArmstrong() which will calculate the given number is armStrong or not. Where readNo() will be private method.
d.	Write a program to demonstrate function definition outside class and accessing class members in function definition.
<b>2.</b>	<b>Using friend functions.</b>
a.	Write a friend function for adding the two complex numbers, using a single class
b.	Write a friend function for adding the two different distances and display its sum, using two classes.
c.	Write a friend function for adding the two matrix from two different classes and display its sum.
<b>3.</b>	<b>Constructors and method overloading.</b>
a.	Design a class Complex for adding the two complex numbers and also show the use of constructor.
b.	Design a class Geometry containing the methods area() and volume() and also overload the area() function .
c.	Design a class StaticDemo to show the implementation of static variable and static function.
<b>4.</b>	<b>Operator Overloading</b>
a.	Overload the operator unary(-) for demonstrating operator overloading.
b.	Overload the operator + for adding the timings of two clocks, And also pass objects as an argument.
c.	Overload the + for concatenating the two strings. For e.g “Py” + “thon” = Python
<b>5.</b>	<b>Inheritance</b>
a.	Design a class for single level inheritance using public and private type derivation.
b.	Design a class for multiple inheritance.
c.	Implement the hierarchical inheritance.

<b>6.</b>	<b>Virtual functions and abstract classes</b>
a.	Implement the concept of method overriding.
b.	Show the use of virtual function
c.	Show the implementation of abstract class.
<b>7.</b>	<b>String handling</b>
a.	String operations for string length , string concatenation
b.	String operations for string reverse, string comparison,
c.	Console formatting functions.
<b>8.</b>	<b>Exception handling</b>
a.	Show the implementation of exception handling
b.	Show the implementation for exception handling for strings
c.	Show the implementation of exception handling for using the pointers.
<b>9.</b>	<b>File handling</b>
a.	Design a class FileDemo open a file in read mode and display the total number of words and lines in the file.
b.	Design a class to handle multiple files and file operations
c.	Design a editor for appending and editing the files
<b>10.</b>	<b>Templates</b>
a.	Show the implementation for the following
b.	Show the implementation of template class library for swap function.
c.	Design the template class library for sorting ascending to descending and vice-versa