LAB MANUAL

FOR

B.TECH PROGRAM

NATIONAL INSTITUTE OF TECHNOLOGY

KURUKSHETRA

LAB MANUAL

FOR

B.TECH PROGRAM

(1st year)

NATIONAL INSTITUTE OF TECHNOLOGY KURUKSHETRA

Lab Manual
Object Oriented Programming using C++ (Pr.)
CSPC-14/ITPC-14

Object Oriented Programming using C++ (Pr.)
CSPC-14/ITPC-14

L	T	P		
-	-	2	Practical:	40
			Sessional:	60

List of Practical

1. (a) Model a geometric point to find distance between two points.
   (b) Model complex numbers and their operations.

2. Describe a class called TOLL- BOOTH with the following data items unsigned int - to hold the number of cars passing through the booth,double - to hold the total amount collected.
   Include the following member functions:
   - a constructor that sets both the data fields to zero.
   - PAYINGCAR( ) that increases the numbers of cars by one andincrease the total amount by 2.50.
   - NOPAYING( ) that increases the number of cars but keeps the total amountunchanged.
   - DISPLAY( ) that displays both the total number of cars passing and the total number of amount collected.

   Write main( ) to test the class thoroughly.

3. Create a class rational, which represents a numerical value by two doublevalues- NUMERATOR and DENOMIATOR . Include the following public memberfunctions:
   - constructor with no arguments (default)
   - constructor with two arguments.
   - void reduce () that reduces the rational number by eliminating the highest common factor between the numerator and denominator.
   - overload + operator to add two rational numbers.
   - overload >> operator to enable input through cin.
   - overload << operator to enable output through cout.

   Write a main () to test all the functions in the class.

4. Consider the following class
   definition class father {
   protected : int age;

   public;
   father (int x){age =x;}
   virtual void iam()

{cout <<"I AM THE FATHER, my age is :"<<age<<end1;}
};

Derive the two classes son and daughter from the above class and for each, define iam() to write out similar but appropriate messages. You should also define suitable constructors for these classes.

Now, write a main () that creates objects of the three classes and then calls iam() for them. Declare pointer to father. Successively, assign addresses of objects of the two derived classes to this pointer and in each case, call iam() through the pointer to demonstrate polymorphism in action.

5. A thermostat is a device that keeps a system at a constants temperature. It behaves like a temperature gauge that is capable of getting the current temperature from the system. It is also a switch that can be turned "on" and "off". The thermostat monitors the temp. in the following manner :

   if the current temp. falls below 95% of the required temp., it turns itself "on". On the other hand, if the current temp. exceeds 1.05 of the required temp. ,it turns itself "off" .In all other cases ,its on-off status remain unchanged.

   Implement classes for temp. gauge and switch(named switch) with suitable data and member functions. The temp. gauge class must have a member function get_temp() that will pretend to get the current temp. of the system by actually reading it from the keyboard.

   Now, implement thermostat class in both the following ways:

   a) Develop a class called thermostat that include objects of temperature gauge and switch as its member (aggregation).
   b) Develop a class called thermostat that inherits the data functions of temp. gauge and Switch (multiple inheritances).

      Write main ( ) to test all the features of above-mentioned classes.

6. Write a program that creates a binary file by reading the data for the students from the terminal. The data of each student consist of roll no., name (a string of 30 or lesser no. of characters) and marks.

7. Using the file created in problem 6, write a program to display the roll no. and names of the students who have passed (has obtained 50 or more).

8. You are to create a file containing n records. Each record relates to a historical event and the year in which the event took place. Some examples are:

   India Wins Freedom 1947 Amartya Sen Gets Nobel 1998 First World War Begins 1914

   The data should be read from terminal while creating the file.

9. A hospital wants to create a database regarding its indoor patients. The information to store include

   (a) Name of the patient
   (b) Date of admission
   (c) Disease
   (d) Date of discharge

   Create a structure to store the date (year, month and date as its members). Create a base class to store the above information. The member function should include functions to enter information and display a list of all the patients in the databases. Create a derived class to store the age of the patients. List the information about all the pediatric patients (less than twelve years in age).

10. Define a class to store the time at a point. The data members should include hr., min., and sec. to store hours, minutes and seconds. The member functions should include functions for reading the time anddisplaying the same. Add a friend function to add two times. Write a program, using the above declaration, to read two times and add them.

11. Write a program to read two matrices and find their product. Use operator overloading so that the statement for multiplying the matrices may be written as $Z \quad X * Y$ where X , Y and Z are matrices.

12. Write a program to read a number and display its square, square root, cube and cube root. Use a virtual function to display any one of the above.

13. Make a class **Employee** with a name and salary. Make a class **Manager** inherit from**Employee**.Add an instance variable, named department, of type String. Supply a method to **toString** that prints the manager's name, department and salary. Make a class **Executive** inherit from **Manager**. Supply a method **toString** that prints the string **"Executive"** followed by the information stored in the **Manager** superclass object. Supply a test program that tests these classes and functions.

14. Write a superclass **Worker** and subclass **HourlyWorker** and **SalariedWorker**. Every worker has a name and a salary rate. Write a method **computePay(int hours)** that computes the weekly pay for every worker. An hourly worker gets paid the hourly wage for the actual number of hours worked, if **hours** is at most 40. If the hourly worker worked more than 40 hours, the excess is paid at time and ahalf. The salaried worker gets paid the hourly wage for 40 hours, no matter what the actual number of hours is. Write a static method that uses polymorphism to compute the pay of any **Worker.** Supply a test program that tests these classes and functions.

Course Outcome (COs)	Description
CO1	To be able to apply an object oriented approach to programming and identify potential benefits of object-oriented programming over other approaches.
CO2	To be able to reuse the code and write the classes which work like built-in types.
CO3	To be able to design applications which are easier to debug, maintain and extend.
CO4	To be able to apply object-oriented concepts in real world applications.

Course Outcomes (COs)	Programme Outcomes (POs)											
	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**
CO1	3	3	2		2							1
CO2	1	2	2	3		1					1	
CO3		3	3		3					1	1	
CO4	2	2	1	2	2	1			1	1		1

Object Oriented Programming using C++ (ITPC-14/CSPC-14) lab Manual

Experiment-1

1. WAP to find out greatest number among three numbers.
2. WAP to calculate the factorial of a given number.
3. WAP to find out the sum of first n integer numbers.
4. WAP to find out the sum of digits of a number.
5. WAP to print 'A' in a matrix with n and m order.
6. WAP to print the following pattern

   A
   A B
   A B C
   A B C D
   A B C D E

Experiment-2

1. WAP to find out whether a given number is prime or not.
2. WAP to print the Fibonacci series.
3. WAP to print the reverse of a number.
4. WAP to check whether a given number is palindrome or not.
5. WAP to check whether a given number is Armstrong number or not.
6. WAP to print the ASCII value of a given character.
7. WAP to swap the values of two variables without using third variable and any arithmetic operator.

Experiment-3

1. Write a program to implement a function that receives a positive floating point number and rounds it to two decimal places. For example 127.565031 rounds to 127.570000. Print the rounded number to six decimal places.
2. WAP to generate the random number from following set without using any conditional statement.

   1, 3, 9, 27, 81, 243, 729, 2187
3. WAP that reads a floating-point number and prints the ceiling, floor, and rounded value.
4. Prepare a payroll earnings statement for the sales force at the Arctic Ice Company. All of Arctic's employees are on a straight commission basis of 12.5 % of sales. Each month, they also receive a bonus that varies depending on the profit of the month and their length of service. The sales manager calculates the bonus separately and enters it with the salesperson's total sales for the month. Your program is also to calculate the withholding taxes and retirement for the month based on the following rates:
   a. Federal withholding: 25%

b. State withholding: 10%
c. Retirement plan: 8%
d. The test data to use for the program are shown in following table.

SALESPERSON	SALES	BONUS
1	53500	425
2	41300	300
3	56800	350
4	36200	175

5. Write a program that asks the user to enter the current date and a person's birth date in the form month, day, year. The program then calculates the person's age in integral years. Use separate functions to enter the dates (pass by address), calculate the person's age, and print the results. Test your program with the following dates: 11/14/1957, 5/10/1989, and 1/5/2000.

Experiment 4

1. We have two arrays, A and B, each of 10 integers. Write a program that tests if every element of array A is equal to its corresponding element in array B. In other words, the function must check if A [0] is equal to B[0], A[1] is equal to B[1], and so forth.
2. Write a function that reverses the elements of an array so that the last element becomes the first, the second from the last becomes second, and so forth. The function is to reverse the elements in place—that is, without using another array.
3. Write a program that sorts a 50-element array using the selection sort, the bubble sort, and the insertion sort. Each sort is to be executed twice.
   a) For the first sort, fill the array with random numbers between 1 and 1,000.
   b) For the second sort, create a nearly ordered list by exchanging every 10^{th} element with its predecessor (9 and 10, 19 and 20, etc.).
   c) Each sort is to sort the same data. For each sort, count the number of comparisons necessary to order the list.
   d) After each sort execution, print the unsorted data followed by the sort data in 5×10 matrixes. After the sorted data, print the number of comparisons and the number of moves required to order the data. Provide appropriate headings for each printout.
   e) To make sure your statistics are as accurate as possible, you must analyze each loop limit condition test and each selection statement in your sort functions.
   f) Analyze the heuristics you generated and write a few lines concerning what you discovered about these sorts.

Experiment-5

1. Write a function using pointers that given the time in seconds passes back the time in hours, minutes, seconds and a character indicating A.M. or P.M. If the number of seconds is more than 24 hours, the function is to return false as an error indicator.
2. Write a function that reverses the elements of an array so that the last element becomes the first, the second from the last becomes second, and so forth. The function must accept only one pointer value and return void.
3. Given the following declaration and definition
   int num[20];
   and using only pointer notation, write a for loop to read integer values from the keyboard to fill the array.
4. Write a program that will read 10 integers from the keyboard and place them in array. The program then will sort the array into ascending order and descending order and print the sorted lists. The program must not change the original array or create any other integer arrays. The solution to this problem requires two pointer arrays. The first pointer array is rearranged so that it points to the data in ascending sequence. The second pointer array is rearranged to that it points to the data in descending sequence.

Experiment-6

1. Write a program to implement a binary member function to subtract one fraction from another. The function should simulate the subtract/assign operator (fr1 - =fr2) and should return void. Fraction is a name of a class that contains two private members numerator and denominator.
2. Write a program to implement a binary member function to multiply two fractions. The function should simulate the multiply/assign operator (fr1 * =fr2) and should return void. Fraction is a name of a class that contains two private members numerator and denominator.
3. Write a program to implement a binary friend function to divide two fractions. The function should simulate the divide/assign operator (fr1 / =fr2) and should return void. Fraction is a name of a class that contains two private members numerator and denominator.
4. Define a class called Array. The class simulates a dynamic array of integers. The class should have two data members. The first data member is the length of the array- i.e. the number of elements in it. The second data member is a pointer to an array that holds the data values.
   The array should have two private member functions: one that extends the array when an element is added and one that contracts it when an element is deleted.
   It should have the following public functions:
   a) It should have one constructor that initializes the pointer to 0.
   b) It should have one logical copy constructor that copies an array.

c) It should have one destructor that destroys the array. The destructor must delete the dynamic memory array.
d) It should have one function that appends one integer at the end of the array.
e) It should have one function that chops the array by deleting the last element.
f) It should have one function that prints the values in the array.

All functions should return a Boolean value: true for success and false for error.

Experiment-7

1. Overload the > operator for the fraction class. The operator should be a binary friend function. It should return a Boolean value.
2. Overload the && as the binary friend operator for the fraction class to determine if neither of the fraction is zero. It should return Boolean.
3. Overload the () operator for the fraction class to extract the integral part of a fraction. It should return an integer. For example, if the fraction is 18/5, it returns 3.
4. Overload the [] operator for the fraction class to extract the fractional part of a fraction. It should return a fraction. For example, if the fraction is 18/5, it returns 3/5.

Experiment-8

1. Define an abstract class called Length that defines the length of an object in millimeters. Then define two classes inherited from Length, called Metric Lenth and English Length. The Metric Length has methods that provide the length of the object in millimeters, centimeters, and meters. The English Length has methods that provide the length of the object in inches, feet and yards.
2. Write a template class, set, that implements a set. A set is an unordered collection of zero or more elements with no duplicates. The public functions are to be:
   a) Constructor
   b) Destructor
   c) Add an element
   d) List the elements
   e) Intersection. An intersection of two sets is another se that contains the common elements from the two sets.

f) Union. A union of two sets is another set that contains the elements in either the first set or the second.

Experiment-9

1. Write a program to implement a function that accepts a C++ string (by reference) and deletes the last character.
2. Write a program to implement a function that accepts a C++ string (by reference) and deletes all the trailing spaces at the end of the string.
3. Write a program to implement a function that returns the number of times a character is found in a C++ string. The function has two parameters. The first parameter is a reference to a string. The second parameter is the character to be counted.
4. Write a program to implement a function that allocates memory for a single data type passed as a parameter. The function uses new operator and returns a pointer to the allocated memory. The function, however, catches and handles any exceptions detected during allocation.
5. Write a program to implement a function that allocates memory for an array given the type of the array and its size as parameters. The function uses the new operator and returns a pointer to the allocated memory. The function, however, catches and handles any exceptions detected during allocation.