2/4 B.Tech. FIRST SEMESTER

CS3L1 OBJECT ORIENTED PROGRAMMING (C++) Lab Credits: 4

(Common to CSE, IT)

Required

Lecture: -- **Internal assessment: 25 marks**

Lab: 6 periods/week **Semester end examination: 50 marks**

Course context and Overview: This lab course provides in-depth coverage of object-oriented programming principles and techniques using C++. Topics include classes, overloading, data abstraction, information hiding, encapsulation, inheritance, polymorphism, file processing, templates, exceptions, container classes, and low-level language features.

Prerequisites: Object Oriented Programming

Objectives:

The objectives of the course are to have students identify and practice the object-oriented programming concepts and techniques, practice the use of C++ classes and class libraries, arrays, vectors, inheritance and file I/O stream concepts.

Learning Outcomes:

Ability to:

1. Creating simple programs using classes and objects in C++.
2. Implement Object Oriented Programming Concepts in C++.
3. Develop applications using stream I/O and file I/O.
4. Implement simple graphical user interfaces.
5. Implement Object Oriented Programs using templates and exceptional handling concepts.

Exercise 1

a) Write a C++ program to display “Hello, Welcome to C++ Programing”
b) Write a C++ program to print details name, rollnumber in a single and two lines.
c) Write a C++ program to print name by reading, assigning and initializing to a variable with an appropriate prompt.
d) Write a C++ program to print your personal details name, surname (single character), totalmarks, gender(M/F), result(P/F) by taking input from the user.

Exercise 2

a) Write a C++ program to convert centigrade into Fahrenheit. Formula: C=(F-32)/1.8
b) Write a C++ program that declares two integers, determines whether the first is a multiple of the second and print the result. (Hint: Use the remainder operator)
c) Write a C++ program that prompts the user to enter two integer values in int variables val1, val2 and find largest, sum, difference, product and ratio of these values.

d) Write a C++ program that prompts the user to enter three integer values, and then outputs the values in numerical sequence separated by commas. So, if the user enters the values 10 4 6, the output should be 4,6,10. If two values are the same, they should just be ordered together. So, the input 4 5 4 should give 4,4,5.

Exercise 3

a) Write a C++ program to read a sequence of double values into a vector. Think of each value as the distance between two cities along a given route. Compute and print the total distance. Find and print the smallest and greatest distance between two neighboring cities. Find and print the mean distance of the neighboring cities.

b) Write a C++ program to convert decimal to binary.

c) Write a C++ program to print the accepted number and its reverse number.

d) Write a C++ program to read marks in 6 subjects using vectors and find average marks

Exercise 4

a) Write a C++ program to read names using vector and display the names and their count.

b) Write a C++ program to read a number between 1 and 100 and the program asks questions to figure out what the number is (e.g. "Is the number you are thinking of less than 50?"). Your program should be able to identify the number after asking no more than seven questions. Hint: use the < and <= operators and the if-else statement.

c) Develop a simple calculator using if-else if and switch-case.

Exercise 5

a) Write a C++ program to find the sum of individual digits of a positive integer.

b) Make a vector holding the ten strings "zero", "one", … , "nine". Use that in a program that converts a digits to its corresponding spelled out values; e.g. the input 5 gives the output five. Have the same program, using the same input loop, convert spelled out numbers into their digit form; e.g., the input five gives the output 5.

c) Write a C++ program to find all the prime numbers between 1 and 100. Write a function to do this.

Exercise 6

a) Write a C++ program that uses functions.

a) to swap two integers b) to swap characters c) to swap two reals

b) Write a C++ program that reads a series of numbers and stores them in a vector<int>. After the user inputs all the numbers he wishes to, ask how many of the numbers the user wants to sum. For an answer N, print the sum of the first N elements of the vector. For example "Please enter some numbers (press '0' at prompt to stop):"

12 23 13 24 15

"Please enter how many of the numbers you wish to sum, starting from the first:"

3

"The sum of the first 3 numbers: 12 23 and 13 is 48"

c) Write a C++ program that writes out the Fibonacci series. Find the largest Fibonacci number that fits in an int.

Exercise 7

a) Define a class name_value that holds a string and a value. Give it a constructor (a bit like Token). Use vector<name_value> instead of two vectors.

b) Create a class employee that includes firstname(type String), lastname(type String) and a monthly salary. Create two employee objects and display each object's yearly salary. Give each employee a 10% raise and display each employee's yearly salary.

Exercise 8

a) Write a C++ program that reads digits and computes them into integers. For example 123 is read as the characters 1,2 and 3. The program should output "123 is 1 hundred and 2 tens and 3 ones". The number should be output as an int value. Handle numbers with one, two, three or four digits. Hint: to get the integer value 5 of the character '5' subtract '0' that is '5'-'0'==5.

b) Provide name constants that you really can't change the value of. Hint: you have to add a member to variable that distinguishes between constants and variables and check for it in set_value(). If you want to let the user define constants. You'll have to add a notation to let the user express that, for example, const pi=3.14.

Exercise 9

a) Write a function print() that prints a vector of ints to cout. Give it two arguments; a string for "labeling" the output and a vector.

b) Write two functions that reverse the order of elements in a vector<int>. The first reverse function should produce a new vector with the reversed sequence, leaving its original vector unchanged. The other reverse function should reverse the elements of its vector without using any other vectors.

Exercise 10

a) Write a function randint() that produces a pseudo-random number in the range [0:MAXINT].

b) Write a function that using randint() from the previous exercise. Computes a pseudo-random integer in the range(a:b). rand_in_range(int a, int b).

c) Write a function that finds the smallest and the largest element of a vector argument and also computes the mean and the median. Do not use global variables. Either return a struct containing the results or pass them back through reference arguments.

Exercise 11

a) Write a function that takes a vector<string> argument and returns a vector<int> containing the number of characters in each string. Also find the longest and the shortest string and the lexicographically first and last string.

b) Write a function that given two vector<double> price and weight computes a value (an "index") that is the sum of all price[i]*weight[i]. Note that we must have weight.size()<=price.size().

Exercise 12

a) Write a C++ program to display the contents of a text file.

b) Write a C++ program that counts the characters, lines and words in the text file.

c) Write a C++ program that produces the sum of all the numbers in a file of whitespace separated integers.

Exercise 13

a) Write a C++ program that creates a file of data in the form of the temperature. Fill the file with at least 50 temperature readings. Call this program store_temps.cpp and the file it creates raw_temps.txt.

b) Write a C++ program that accepts two file names and produces a new file that is the contents of the first file followed by the contents of the second; that is, the program concatenates the two files.

Exercise 14

a) Write a C++ program that given a file name and a word outputs each line that contains that word together with the line number. Hint:getline().

b) Write a C++ program that reads a text file and converts its input to all lower case, producing a new file.

Exercise 15

1. Write a C++ program that removes all vowels from a file. For example, once upon a time! Becomes nc pn tm!.
2. Write a C++ program that replaces punctuation with whitespace. For example, "don't use the as-if rule" becomes dont use the asif rule".
3. Write a C++ program to reverse the order of characters in a text file. For example, asdfghjkl becomes lkjhgfdsa.

Exercise 16

a) Write a C++ program that reads a text file and writes out how many characters of each character classification are in the file.

b) Write a C++ program draw a rectangle as a rectangle and as a polygon. Make the lines of the polygon red and the lines of the rectangle blue.

c) Write a C++ program draw a 100-by-30 rectangle and place the text "PVPSIT" inside it.

Exercise 17

a) Write a C++ program to draw the Olympic five rings.
b) Write a C++ program to display an image like photo on the screen.
c) Write a C++ program to draw a part of an ellipse by defining a class arc. Hint:fl_arc().

Exercise 18

a) Write a C++ program to draw a box with rounded corners. Define a class box, consisting of four lines and four arcs.
b) Write a C++ program to draw a line with an arrowhead by defining a class arrow.

Exercise 19

a) Define a class poly that represents a polygon but checks that its points really do make a polygon in its constructor. Hint: you'll have to supply the points to the constructor.
b) Define a class star. One parameter should be the number of points. Write a C++ program to draw a few stars with differing numbers of points, differing line colors, and differing fill colors.
c) Define two classes smiley and frowny, which are both derived from class circle and have two eyes and a mouth. Next, derive classes from smiley and frowny, which add an appropriate hat to each.

Exercise 20

a) Write a C++ program to write a function void to_lower(char* s) that replaces all uppercase characters. Don't use any standard library functions.
b) Write a C++ program to write a function, char* findx(const char* s, const char* x), that finds the first occurrence of the string x in s.

Exercise 21

a) Write a C++ program that reads characters from cin into an array that you allocate on the free store. Read individual characters until an asterisk (*) is entered. Do not use a std::string.
b) Write a C++ program to write a function, char* strdup(const char*) that copies a string into memory it allocates on the free store. Use the dereference operator * instead.

Exercise 22

a) Write a C++ program to write a function char* findx(const char* s, const char* x) that find the first occurrence of the string x in s. Use dereference operator * instead.
b) Write a C++ program to write a function string cat_dot(const string& s1, const string& s2), that concatenates two strings with a dot in between.
c) Write a template function that adds a vector of elements of an object of any type to which elements can be added.

Exercise 23

a) Write a C++ program to write a function template for finding the minimum value contained in any array.

b) Write a template function that takes a vector<T>vt and a vector<U> vu as arguments and returns the sum of all vt[i]*vt[i]s.

c) Define a class Int having a single member of class int. Define constructors, assignment, and operators +,-,*,/ for it.

d) Implement vector::operator=() using an allocator for memory management.

Exercise 24

a) Define a file_handle class with a constructor that takes a string argument (filename), opens the file in the constructor, and closes it in the destructor.

b) Define an input and an output operator (>> and <<) for vector.

c) Given a list<int> as a (by-reference) parameter, make a vector<double> and copy the elements of the list into it. Verify that the copy was complete and correct. Then print the elements sorted in order of increasing value.

Exercise 25

a) Define a singly-linked list, slist, and perform operations insertion, deletion and traverse.

b) Define a pvector to be like a vector of pointers except that it contains pointers to objects and its destructor deletes each object.

Exercise 26

Case study