*Look forward,
learn modern
knowledge, and
do not waste
time in studies
of old subjects of
no values.*

2019

Computer Lab Manual

M.C.A.
Semester-I

CSM-1171: LABORATORY COURSE –I

DEPARTMENT OF COMPUTER SCIENCE

ALIGARH MUSLIM UNIVERSITY ALIGARH

Credits

Lab Manual Design Committee:

➪ *Prof. Mohammad Ubaidullah Bokhari*

➪ *Dr. Arman Rasool Faridi*

➪ *Dr. Faisal Anwar*

➪ *Dr. Aasim Zafar (Convener)*

Editor:

➪ *Dr. Aasim Zafar*

Design & Compilation:

➪ *Mr. S. K. Sharma*

Third Edition: ***July 2019***

Approved by BoS, Dated-……………………

Department of Computer Science
A.M.U., Aligarh, (U.P.), India

Lab Manual: Laboratory Course – I (CSM-1171)

LIST OF CONTENTS (WEEK WISE)

COURSE TITLE: LABORATORY COURSE-I	**COURSE CODE:** CSM – 1171
CREDIT: 4	**PERIODS PER WEEK:** 6
CONTINUOUS ASSESSMENT: 40	**EXAMS:** 60

COURSE DESCRIPTION

This course examines practical programming techniques and issues, emphasizing object modelling and simulation. The objectives of the course are to explore issues involved in developing large-scale object-oriented systems and to teach fundamental techniques that can simplify software development. The course provides in depth information on object oriented programming, issues, techniques, and methodologies.

This course also provides the practical knowledge of system analysis and design. System analysis and design deal with planning the development of information systems through understanding and specifying in detail what a system should do and how the components of the system should be implemented and work together.

CONTENT

This course introduces to fundamental ‘computer literacy’ concepts. The objective is to explore the knowledge of fundamental of information technology and information system.

This course is designed to provide the students the opportunity of learning both – concepts of C++ and then implementing algorithms and data structures. This course is indented to develop a deep understanding of various operations on data structure such as searching, sorting, insertion, deletion and traversing.

This course also deals with the concepts, skills, methodologies, techniques, tools, and perspectives essential for systems analysts. System analysts solve business problems through analyzing the requirements of information systems and designing such systems by applying analysis and design techniques.

OBJECTIVES

This course is designed to help students in:

- ❒ To help the students in learning the C++.
- ❒ To help the students how to write a sample program in C++.
- ❒ To help the students in understanding the use of different variables.
- ❒ To help the students in understanding the use of decision making statement.
- ❒ Be able to debug and test C++ programs;
- ❒ Understand how to read C++ library documentation & reuse library code.
- ❒ To make the students understand the features of object oriented principles and familiarize them with virtual functions, templates and exception handling.
- ❒ Design ER diagram.
- ❒ Design Data Flow diagram of various levels.

OUTCOMES

After completing this course, the students would be able to:

- ❒ Write, debug and run a sample program in C++.
- ❒ Differentiate the variables and constants.
- ❒ Understand how decision making statements are written.

- ❒ Create a simple real life applications based on the decision making statements.
- ❒ Understand the concepts and implementation of constructors and destructors.
- ❒ Develop software applications using object oriented programming language in C++
- ❒ Understand and use the basic programming constructs of C++
- ❒ Learn C++ data types, memory allocation/reallocations, functions and pointers.
- ❒ Apply object oriented programming concepts to software problems in C++
- ❒ Understand the concept of System Analysis and Design.
- ❒ Create the E-R Diagrams of any real life problems.

RULES AND REGULATIONS

Students are required to strictly adhere to the following rules.

- ❒ The students must complete the weekly activities/assignments well in time (i.e., within the same week).
- ❒ The students must maintain the Lab File of their completed activities/assignments in the prescribed format (**Appendix-1**).
- ❒ The students must get the completed weekly activities/assignments checked and signed by the concerned teachers in the Lab in the immediate succeeding week. Failing which the activities/assignments for that week will be treated as incomplete.

- ❒ At least **TEN (10)** such timely completed and duly signed weekly activities/assignments are compulsory, failing which students will not be allowed to appear in the final Lab Examination.
- ❒ The students need to submit the following three deliverables for each exercise duly signed by the Teacher:
  - Algorithm
  - Flow Chart
  - Coding
  - Input /Output
- ❒ Each question will be evaluated on a scale of 10 points, 4 for Algorithm, 3 for Flow Chart and 3 for Coding part.
- ❒ The students need to ensure that each question is assessed and signed by the Teacher in the week/time.
- ❒ Late submission would not be accepted after the due date.
- ❒ Cooperate, collaborate and explore for the best individual learning outcomes but copying is strictly prohibited.

APPENDIX–I

Template for the Index of Lab File

WEEK NO.	PROBLEMS WITH DESCRIPTION		PAGE NO.	SIGNATURE OF THE TEACHER WITH DATE
1	1#			
	2#			
	3#			
2	1#			
	2#			
	3#			
3	1#			
	2#			
	3#			

Note: The students should use Header and Footer mentioning their roll no. & name in footer and page no in header.

WEEK #1

OBJECTIVES

- To help the students in learning *the practical use of MS-Word, MS-Excel, MS-Access, MS-Power Point & MS-Front Page tool.*
- *To learn using HTML Tags.*
- To help the students in learning *the Practical use of HTML.*

OUTCOMES

After completing this, the students would be able to:

- Use MS-Word, MS-Excel, MS-Access, MS-Power Point & MS-Front Page in real life applications.
- Create a simple web site using HTML Tags.

PROBLEMS

1# Open a new document and type the following letter.

July 15, 2016,
Chennai.

From
VENKATESH .P
Sri Ranga Apartments,
No: 120, II Avenue,
T. Nagar. Chennai-17
To

<<Name>>
<<Address>>

Respected <<Name>>

With the current slowdown in hiring within the high- tech field, you must be flooded with resumes from out-placed software engineers such as me. Please take a moment to consider my qualifications. I believe in particular is highly marketable in this tight market:

I worked on the team that pioneered the technology that put the Palm Pilot on the map.

In today's increasingly mobile society, this technology has places to go, and I have ideas that could take us to the next step in office independence.

Please call me with prospective job opportunities. I am interested in a project management position in the Rs. 9K range.

Thank you!
Venkatesh.

Enclosure: Resume (the format given below).

i) Save the document as "Letter.doc."

ii) Send the document to 3 recipients using Mail merge. (Use 3 different addresses)

iii) Define a Macro 'Decorate' which makes the text bold, Red in color and italic, font size Assign a shortcut key Alt + Z to this macro.

iv) Close the document.

v) The Sample Addresses are:

   i) Mr. Amit Tandon

13, New Estate,
Ring Road, Chandigarh

ii) Mr. Rohit Saluja
15, Karol Bagh, New Delhi

iii) Ms. Jyoti Parmar
Sector 16, New Building, Gurugram

Format of Resume

Name :

Father's Name :

Date of Birth :

Age :

Address :

Educational Qualification :

Sr.	**Qualification**	**Board/University**	**Percentage**

Work Experience :

Technical Skills :

Personal Skills :

Hobbies :

Dated:…………………. Signature

2# Create a table in word as shown below:

Roll No	**Name**	**Marks in Physics**	**Marks in Chemistry**	**Total Marks**
1	Sakshi	80	70	
2	Rohit	70	80	
3	Amit	60	50	
4	Rakesh	40	60	

5	Komal	30	70	
6	Garima	80	80	

Do the following:

(a) In the total marks column, entries should be calculated using formulas and it is the sum of marks in physics and marks in chemistry.

(b) Insert a new row at the end of the table and also find grand total using formula.

(c) Sort the table based on total marks.

(d) The date and heading should be centre aligned.

(e) Heading should be in bold and underlined.

3# Using a spreadsheet package you have studied, construct T Morongo’s pay slip for December 2016 following the instructions below. Insert a custom footer with your *name*, *subject*, *course*, *exam/Test & question number*. Save it as Salary advice.

SALARY ADVICE FOR MARCH 2016			
EMPLOYEE	T MARONGO		
STAFF NO	004		
DATE	31 MARCH 2016		
NEXT PAY DATE	30 APRIL 2016		
BASIC SALARY p.a.	31200.00		
INCOME	AMOUNT	DEDUCTIONS	AMOUNT
Basic Salary		Pension @8%	
Housing Subsidy		P.A.Y.E.	
Vehicle Allowance		U.I.F.	
		Medical Aid	
		Bond Repayment	

Gross Income		Total Deductions	
Net Salary			

Instructions:

i) Housing Subsidy 6000.00 per year.

ii) Car Allowance 100.00 per month,

iii) Pension 8% on Basic Salary.

iv) P.A.Y.E. 636.83

v) Medical Aid 70.00

vi) U.I.F. 1% on Basic Salary + Housing Subsidy

vii) Bond Repayment 630.00

viii) Calculate Net Salary.

ix) Format all figures to two decimal places and insert ₹ currency symbol.

x) Insert a custom footer with your name, subject, and question number. Save it as salary advice2.

4# Use a new workbook & construct a worksheet with the data given & save it as LYONS.

LYONS INC

Orange JUICE Sales

PRODUCT	COST PRICE PER LITRE	MARK UP PER ITEM 35%	SELLING PRICE	LITRES SOLD	TOTAL INCOME	PROFIT
Cascade	3.75			234		
Quench	3.65			345		
Xtra	4.25			456		
Sun Splash	1.50			123		

House Brand	1.50			245		
TOTAL						
HIGHEST						
LOWEST						

Instructions

- ✓ **Markup** = Cost price/Litre x 35%
- ✓ **Selling price**= Cost price/Litre + Mark up
- ✓ **Total income**= Litres sold x Selling Price
- ✓ **Profit** = Total income – (Cost price/Litre x Litres sold)

a) The MARKUP % (35%) must be inserted in a separate cell under the heading. USE IT as an absolute cell reference in the formula to calculate the markup per item.

b) Calculate the mark up for each item.

c) Calculate the selling price for each item.

d) Calculate the Total Income for each item.

e) Calculate the profit for each item.

f) Format the column LITRES SOLD to display the number of litres as integers. The rest of the worksheet must be formatted to display two decimals.

g) Use statistical functions to calculate the:

- ✓ AVERAGE
- ✓ HIGHEST (MAX)
- ✓ LOWEST (MIN) for Selling Price column up to Profit Column.

h) Show all formulas you have used in a new sheet. Adjust the column width so that the formulae are displayed in full and the sheets fits into one side of A4 landscape format and save it as formulas.

i) Under the worksheet Create a pie chart titled PRODUCT COST PER UNIT for Product & Cost price per Litre columns. Data labels indicating percentages should be displayed.

j) Put borders neatly on the on the work sheet & save it as LYONS2.

5# Create a new database, save it on the desktop and name it "School Database"

i) Create a Table in the *School Database* with the following:

ii) Make the "ID Number" Field as the Primary Key.

iii) Save the table as "Student's Table"

v) Return to the main Access window.

vi) Open the "Students Table" and enter 5 complete records.

vii) Sort the table in ascending order by surname

viii) Move the *Date of Birth* and *Telephone Number* fields so that the *Date of Birth* field is now directly after the *Surname* field.

ix) Delete the last Record you have entered

x) Change the field size of the *Surname* to 20.

xi) Create a Form with all fields on the *Student's* Table.

xi) Name the form *Students Entries*

xii) Make the ID Number of Each student in the form, Red

xiii) Insert a Picture in the form in way that all text is visible.

xiv) Create a report based on the Student's Table showing the Fields *Name, Surname,* and *Telephone Number*.

xv) Name the report *Telephone List*

xvi) Insert a picture in the report Header.

xvii) Create a query, showing all fields of those students who have a particular surname of your Choice.

xviii) Create another query showing all fields of those students born after 1987.

xix) Create a query showing only the Student’s Name, Surname and Date of birth.

Field Name	**Data Type**	**Field Size or Format**
ID Number	Text	10
Name	Text	15
Surname	Text	15
Telephone Number	Number	Long Integer
Date of Birth	Date/Time	Medium Date
Stipend	Currency	Currency
Foreigner	Yes/No	Yes/No

6# Data Entry Form

From the Database Window, make sure the STUDENTS table is selected. From the Insert menu, select AutoForm. A form will be created automatically using the Fields and Records from your table. You can move from one record or Field to another using your mouse or the keyboard. Use any of the above methods to move to a new, blank record. Complete the form. Use the form to add the records 4 to 6 as shown below:

Student Number; Last Name; First Name; Date of Birth; Address; Suburb; Postcode; State; Phone; Gender; Mark; Comment

After you have entered the last record, close the form. You will be prompted to save.

Since this was only a temporary table to use for data entry, click No.

Open the table and you will see the new records listed.

7# Design Seasonal Greeting cards using MS-Power Point.

8# Design a AMU Magazine cover in MS-Power Point. Use the following:

i) Select a theme for the page,

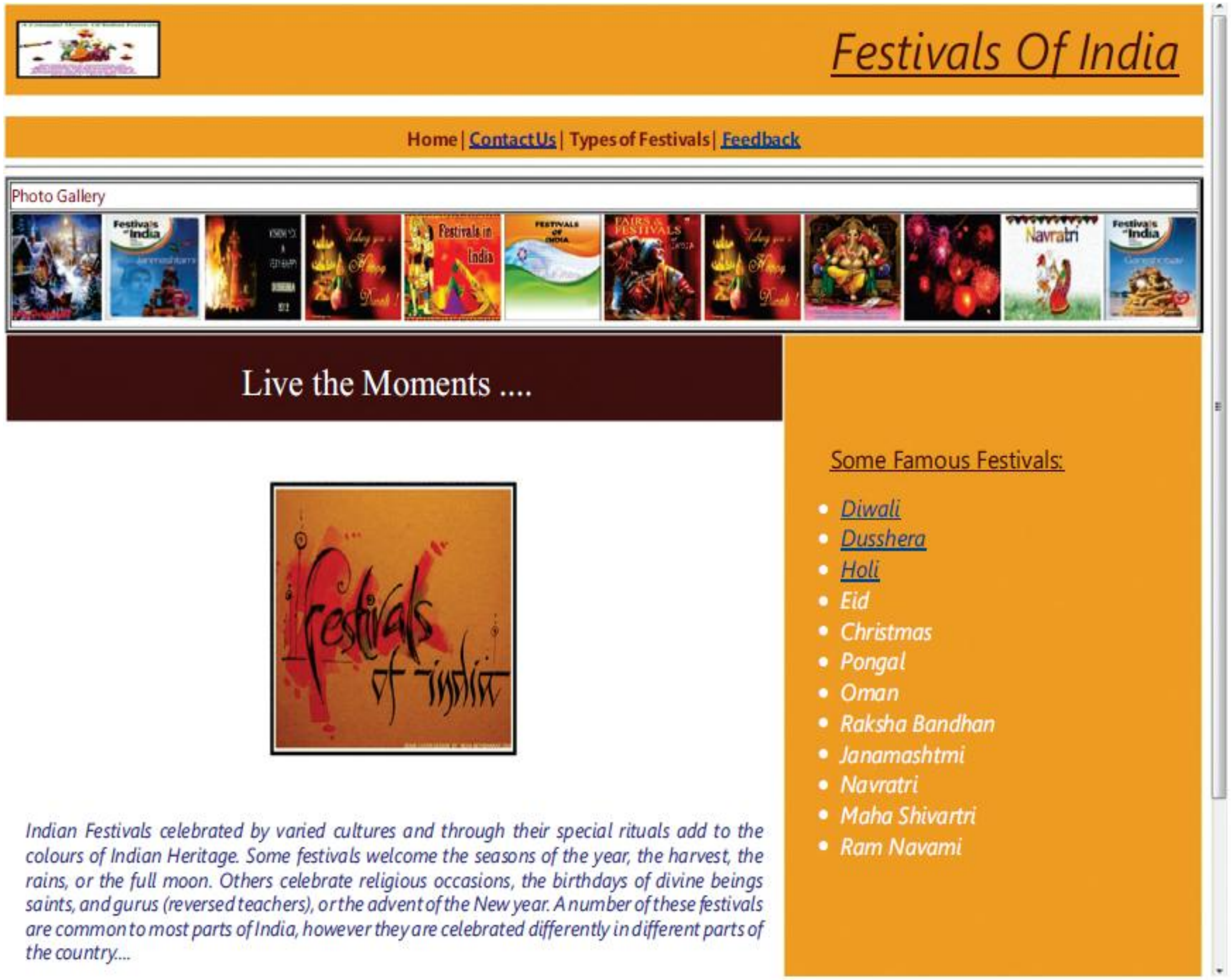
ii) Insert either a picture or clipart, and

iii) Use WordArt.

9# Design a poster inviting all students of your department to the IT Fest (using MS-Power Point).

10# Create a 5-slide presentation on any topic. Use Images, Graphs, Chart, Tables, Animation, Time, Bullets, Transition, Sound, Hyperlink, Background template, Header and Footer (using MS-Power Point).

11# Create a website depicting different festivals of India. The Home page should have links of all festivals of India. It should also offer link to the other web pages and each of these web pages should contain a link to home page and to the various html pages required in your site. Use various effects to the images to be included in the website (using MS-Front Page/HTML Tags)



Festivals Of India

Home | ContactUs | Types of Festivals | Feedback

Photo Gallery

Live the Moments ....

Indian Festivals celebrated by varied cultures and through their special rituals add to the colours of Indian Heritage. Some festivals welcome the seasons of the year, the harvest, the rains, or the full moon. Others celebrate religious occasions, the birthdays of divine beings saints, and gurus (reversed teachers), or the advent of the New year. A number of these festivals are common to most parts of India, however they are celebrated differently in different parts of the country....

Some Famous Festivals:

- Diwali
- Dusshera
- Holi
- Eid
- Christmas
- Pongal
- Oman
- Raksha Bandhan
- Janamashtmi
- Navratri
- Maha Shivartri
- Ram Navami

WEEK #2

OBJECTIVES

- To help the students in learning about operators and learn how to combine different types of arithmetic operators to form arithmetic expressions used in C++.
- To help the students how to combine the increment/decrement and arithmetic operators with the assignment operator to form meaningful expressions in C++.
- To learn concept of Physical DFD, Logical DFD.
- To learn draw concept of level 0 DFD, Level 1DFD, Level 2 DFD.
- To learn how to draw a Data Flow Diagram.

OUTCOMES

After completing this, the students would be able to:

- Understand the usage of pre/post increment/decrement operators
- Understand the precedence of operators and how the precedence rules affect expression evaluation.
- Understand the meaning of type conversion.
- draw Data Flow Diagram of real life applications

PROBLEMS

1# Write a C++ program to check whether a number is even or odd using ternary operator.

2# Write a C++ program to perform the addition of two numbers without using + operator.

3# Write a C++ program to evaluate the arithmetic expression ((a + b / c * d - e) * (f - g)). Read the values a, b, c, d, e, f, g from the standard input device.

4# Write a C++ program to find the sum of individual digits of a 3 digit number.

5# Write a C++ program to read the values of x and y and print the results of the following expressions in one line: (x + y) / (x - y) , (x + y)(x - y)

6# Prepare physical, logical, context level and level 1 DFDs for following activities.

a. Returning a book to a library.
b. Getting a ticket reserved for a train journey.
c. Mess management system.
d. Buying an item from a supermarket.

WEEK #3

OBJECTIVES

- ❒ To help the students in learning the concepts of loops used in C++.
- ❒ To help the students how to write a program using *for* loop in C++.
- ❒ To learn how to draw a Data Flow Diagram.

OUTCOMES

After completing this, the students would be able to:

- ❒ Write programs consisting repeatedly executing same statements in C++.
- ❒ Use the concept of *for* loop in different types of real life and mathematical problems.
- ❒ Draw Data Flow Diagram of real life applications.

PROBLEMS

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

2# A Fibonacci sequence is defined as follows: The first and second terms in the sequence are 0 and 1. Subsequent terms are found by adding the preceding two terms in the sequence. Write a C++ program to generate the first n terms of the sequence.

3# Write a C++ program to generate all the prime numbers between 1 and n, where n is a value supplied by the user.

4# A character is entered through keyboard. Write a C++ program to determine whether the character entered is a capital letter, a small case letter, a digit or a special symbol using if-else and switch case. The following table shows the range of ASCII values for various characters.

Characters ASCII values A – Z: 65 – 90, a – z: 97 – 122, 0 – 9: 48 – 57

Special symbols 0 – 47, 58 – 64, 91 – 96, 123 – 127

5# If cost price and selling price of an item is input through the keyboard, write a program to determine whether the seller has made profit or incurred loss. Write a C++ program to determine how much profit or loss incurred in percentage.

6# Write a C program, which takes two integer operands and one operator from the user, performs the operation and then prints the result. (Consider the operators +, -, *, /, % and use switch statement).

7# Write a C++ program to calculate the following sum:

$$sum = 1 - x^2 /2! + x^4 /4! - x^6 /6! + x^8 /8! - x^{10}/10!$$

8# Write a C++ program to find the roots of a quadratic equation.

9# Write a C++ program to check whether a given 3 digit number is Armstrong number or not.

10# Admission procedure in a university is as follows:

An advertisement is issued giving essential qualifications for the course, the last date for receipt of application, and the fee to be enclosed with the application. A clerk in the registrar's office checks the received application s to see if mark sheet and fee are enclosed and sends valid applications to the concerned academic department. The department checks the application in detail and decides the applicants to be admitted, those to be put in the waiting list, and those to be rejected. Appropriate letters are sent to the Registrar's office which intimates the applicant. **Give all levels of DFDs (0, 1, 2) corresponding to the above problem.**

WEEK #4

OBJECTIVES

- ❒ To help the students in learning the concepts of *arrays* (1D and 2D) in C++.
- ❒ To help the students in implementing and handling 1D and 2D arrays with the help on various examples.
- ❒ To help the students in understanding various operations associated with the arrays.
- ❒ To learn drawing Physical DFD, Logical DFD.

OUTCOMES

After completing this, the students would be able to:

- ❒ Handle arrays (1D and 2D) in the programs.
- ❒ Develop & implement simple real life examples of arrays.
- ❒ Draw the Physical DFD, Logical DFD of real life problems.

PROBLEMS

1# Write a C++ program to find the second largest integer in a list of integers.

2# Write a C++ program to perform the following:

- Addition of two matrices
- Multiplication of two matrices

3# Write a C++ program to count and display positive, negative, odd and even numbers in an array.

4# Write a C++ program to merge two sorted arrays into another array in a sorted order.

5# Write a C++ program to find the frequency of a particular number in a list of integers.

6# A list of employees with their basic pay is sent to a clerk. He calculates the gross pay using standard allowances which are known for each pay slab. Deduction statements such as loan repayment, subscription to association, etc., are also sent to another clerk matches these slips with the slips of gross pay and calculates net pay. This slip is used by a third clerk to write out pay cheques for each employee and sent to respective employees. The total pay bills paid are also computed. **Give physical and logical DFDs corresponding to the above problem.**

WEEK #5

OBJECTIVES

- ❐ To help the students in learning the concepts of *pointers* in C++.
- ❐ To help the students in implementing and handling pointers with the help on various examples.
- ❐ To help the students in understanding various operations associated with the pointers.
- ❐ To learn the Entity Relationship Diagram.

OUTCOMES

After completing this, the students would be able to:

- ❐ Handle pointers in the programs.
- ❐ Develop & implement simple examples of pointers.
- ❐ Implement extensive use of pointers for memory, array, structures and functions.
- ❐ Draw Data Entity Relationship Diagram of real life problems.

PROBLEMS

1# Write a C++ program to concatenate two strings using pointers.

2# Write a program for reading elements using pointer into array and display the values using array.

I. Declare set of elements.

II. Declare pointer and initialize it to first element address of set of elements (array).

III. Repeat loop until pointer reaching to last element and display each element.

3# Write a program through pointer variable to sum of n elements from array.

4# Write a program for reading elements using pointer into array and display the values using array.

5# Write a C++ program to reverse a string using pointers.

6# Give Entity Relationship Diagram corresponding to the given problem.

A small accounting firm wants a simple HR application that will help it to keep track of its employees, their positions, allowances, salary scales, and which company vehicles their employees drive. The application must keep track of all the positions at the firm, the employees filling these positions, the allowances for these positions, the salary scales for these positions, and the company vehicles assigned to these positions.

List of entities and their corresponding attributes:

Employee: Emp_Id (Primary Key), Name, Gender, DoB, Email, Address

Position: Pos_Id (Primary Key), Pos_Name, Pos_Description, Details

Allowances: Allow_Id (primary Key), Allow_Name, Allow_Description, Amount

Salary_Scales: S_Code (Primary Key), S_Name, S_Description, Min_Salary, Max_Salary

Vehicles: Vehl_id(Primary Key), Registration_no, Year, Make, Model

WEEK #6

OBJECTIVES

- ❒ To help the students in implementing and handling pointers with the help on various examples.
- ❒ To help the students in understanding various operations associated with the pointers.
- ❒ To help the students in understanding the concept of *this* pointer,
- ❒ To learn to draw the Entity Relationship Diagram of real life problems.

OUTCOMES

After completing this, the students would be able to:

- ❒ Handle pointers in the programs.
- ❒ Develop & implement simple examples of pointers.
- ❒ Implement extensive use of pointers for memory, array, structures and functions.
- ❒ Design the Entity Relationship Diagram of real life problems.

PROBLEMS

1# Write a C++ program using pointers to compute the sum, mean and standard deviation of all elements stored in an array of **n** real numbers.

2# Write a C++ program to create three objects for a class named pntr_obj with data members such as roll_no & name. Create a member function set_data() for setting the data values and print() member function to print which object has invoked it using *‘this’* pointer

3# Develop a C++ program to find the greatest of two numbers using this pointer which returns the member variable.

4# Write a C++ program to implement flight class with data member as flight_no., source destination and fare. Write a member function to display the flight information using this pointer.

5# Write a C++ program to use *this* pointer and return pointer reference.

6# The registrar at a small college wants an application that will help their department keep track of the schedule of classes, the courses and lecturers appearing in the schedule, and the students registering for courses according to the schedule. Courses are scheduled every semester and this is documented in the schedule of classes, which also documents the lecturers assigned to each schedule of a class. Students register for courses according to the schedule of classes. Users (students, lecturers, and other college staff) must login to the application to gain access, and the application must keep track of user logins/logouts. In addition, users must have different levels of access, which will determine their access to different parts of the application. .

List of entities and their corresponding attributes:

Student: Enrol_No(Primary Key), Name, Gender, DOB, Email, Address, State.

Course: C_Id (Primary Key), C_Name, C_Code, C_Description

Schedule_of_Classes: S_Id (Primary Key), S_Code, Section, day, time, location.

Semester: Sem_Id (Primary Key), Sem_Code, Sem_Name, year, start_date, end_date

Lecturer: L_Id (Primary Key), name, gender, mobile,address, Email, state

Users: User_id (Primary Key), user_name, password

Access_Level: ACL_Id (Primary Key), ACL_Code, Short_Name, Long_Name, ACL_Description

Log_Entries: Log_Id(Primary Key), Logged_On, Logged_Off

Give Entity Relationship Diagram corresponding to the above problem.

WEEK #7

OBJECTIVES

- ❒ To help the students in learning the concepts of *strings* in C++.
- ❒ To help the students in implementing and handling strings operations on string with the help on various examples.

OUTCOMES

After completing this, the students would be able to:

- ❒ Handle strings manipulation in the programs.
- ❒ Develop & implement simple real life examples of strings/string functions.

PROBLEMS

1# Write a C++ program that uses functions to perform the following operations:

   i. To insert a sub string into a given main string from a given position.

   ii. To delete n characters from a given position in a given string.

2# Write a C++ program to determine if the given string is a palindrome or not.

3# Write a C++ program to find a string within a sentence and replace it with another string.

4# Write a C++ program that reads a line of text and counts all occurrence of a particular word.

5# Write a C++ program that displays the position or index in the string S where the string T begins, or 1 if S doesn't contain T.

WEEK #8

OBJECTIVES

- ❒ To help the students in learning the concepts of *functions* in C++.
- ❒ To help the students in implementing and handling about writing your own functions with the help on various examples.
- ❒ To help the students in learning Recursion.
- ❒ To help the students in understanding various operations associated with the function using array.
- ❒ To learn how to design the E-R Diagram of a real life problem.

OUTCOMES

After completing this, the students would be able to:

- ❒ Understand how to access and use library functions in the programs.
- ❒ Understand function declaration and definition.
- ❒ Understand Functions & Recursion.
- ❒ Design the E-R Diagram of a real life problem.

PROBLEMS

1# Write C programs that use both recursive and non-recursive functions to find:

a) The factorial of a given integer.

b) To find the greatest common divisor of two given integers.

2# Write C programs that use both recursive and non-recursive functions to solve towers of Hanoi problem.

3# Write a C++ program to print the transpose of a given matrix using function.

4# Write a C++ program to swap two number by both call by value and call by reference mechanism, using two functions swap_value() and swap_reference respectively, by getting the choice from the user and executing the user's choice by switch-case.

5# Write a C++ program to display all array elements using recursion.

6# Write a C++ program to find sum of elements of array using recursion.

7# Write a C++ program to find maximum and minimum elements in array using recursion.

8# Suppose you are given the following requirements for a simple database for the National Hockey League (NHL):

❒ The NHL has many teams.

❒ Each team has a name, a city, a coach, a captain, and a set of players.

❒ Each player belongs to only one team.

❒ Each player has a name, a position (such as left wing or goalie), a skill level, and a set of injury records.

❒ A team captain is also a player.

❒ A game is played between two teams (referred to as host_team and guest_team) and has a date (such as May 11th, 1999) and a score (such as 4 to 2).

Construct a clean and concise ER diagram for the NHL database using the Chen notation as in your textbook. List your assumptions and clearly indicate the cardinality mappings as well as any role indicators in your ER diagram.

WEEK #9

OBJECTIVES

- ❒ To help the students in learning the concepts of *Structure* in C++.
- ❒ To help the students in implementing *structure* with the help on various examples.

OUTCOMES

After completing this, the students would be able to:

- ❒ Handle problem based on *structure*.
- ❒ Develop & implement simple real life examples of structure.

PROBLEMS

1# Write a C++ program that uses functions to perform the following operations:

i. Reading a complex number

ii. Writing a complex number

iii. Addition and subtraction of two complex numbers

iv. Multiplication of two complex numbers. Note: represent complex number using a structure.

2# Write a C++ program to compute the monthly pay of 100 employees using each employee's name, basic pay. The DA is computed as 52% of the basic pay. Gross-salary (basic pay + DA). Print the employees name and gross salary.

3# Create a Book structure containing book_id, title, author name and price. Write a C++ program to pass a structure as a function argument and print the book details.

4# Create a union containing 6 strings: name, home_address, hostel_address, city, state and zip. Write a C++ program to display your present address.

5# Write a C++ program to define a structure named DOB, which contains name, day, month and year. Using the concept of nested structures display your name and date of birth.

WEEK #10

OBJECTIVES

- ❒ To help the students in learning the concept of OOPs.
- ❒ To help the students how to write a sample program using OOPs.
- ❒ To help the students in understanding the use of class and objects.
- ❒ To help the students in understanding the concept of Access Specifier.
- ❒ To learn how to design the E-R Diagram of a real life problem.

OUTCOMES

After completing this, the students would be able to:

- ❒ Write, debug and run a sample program using OOPs.
- ❒ Create simple real life applications.
- ❒ Understand following points:
  - ✓ Class and object.
  - ✓ Crating class and object.
  - ✓ Defining function in different ways.
- ❒ Concept of Access Specifier like private, public etc.
- ❒ Design the E-R Diagram of a real life problem.

PROBLEMS

1# Write a program in C++ to display your name, Branch, Year on to the computer screen without using classes and object. All information should be displayed in the separate line.

2# Write a menu driven program in C++ to perform all basic arithmetic operation addition, subtraction, multiplication, and division of two given values. Program receives two values and required operation to be performed from the keyboard and display particular result of the required operation.

3# Write a menu driven program in C++ that receives 4 digit integer value the keyboard and perform following operations:

i. Reverse of that no.

ii. sum of number with it’s reverse

iii. sum of alternative digits(1 digit+3 digit and 2 digit+4 digit)

4# Write a menu driven program in C++ to receive integer number and convert equivalent binary, octal, hexadecimal number.

5# Write a menu driven program in C++ to perform all basic arithmetic operation addition, subtraction, multiplication, and division of two given values using function and switch case. Program receives two values and required operation to be performed from the keyboard and display particular result of the required operation.

6# **Estate Agency case study:** Clients wishing to put their property on the market visit the estate agent, who will take details of their house, flat or bungalow and enter them on a card which is filed according to the area, price range and type of property. Potential buyers complete a similar type of card which is filed by buyer name in an A4 binder. Weekly, the estate agent matches the potential buyer's requirements with the available properties and sends them the details of selected properties. When a sale is completed, the buyer confirms that the contracts have been exchanged, client details are removed from the property file, and an invoice is sent to the client. The client receives the top copy of a three part set, with the other two copies being filed. On receipt of the payment the invoice copies are stamped and archived. Invoices are checked on a monthly basis and for those accounts not settled within two months a reminder (the third copy of the invoice) is sent to the client.

Create a context diagram and Ist level DFD for this Estate Agency case study.

WEEK #11

OBJECTIVES

- ❐ To help the students in learning the concept of OOPs.
- ❐ To help the students how to write a sample program using OOPs.
- ❐ To help the students in understanding the use of class and objects.
- ❐ To help the students in understanding the concept of arrays of Objects.

OUTCOMES

After completing this, the students would be able to:

- ❐ Write, debug and run a sample program using OOPs.
- ❐ Create simple real life applications.
- ❐ Understand following points:
  - ✓ Class and object.
  - ✓ Crating class and object.
  - ✓ Defining function in different ways.
- ❐ Concept of arrays of Objects.

PROBLEMS

1# Write a program in C++ to display mark sheet of the student. Define a class that contains data members to store student information line name, Branch, semester, marks in 6 different subjects, etc. Declare some member functions to get this information from the key board, to calculate result and to display all gathered information on to the computer screen in proper format.

2# Define a class employee. Include the following members:

Data Members:

- Name of the employee
- Age of the employee

Member Functions:

- To get the name and age of the employee
- To display the name and age of the employee

3# Define a class Bank Account to represent a bank account. Include the following members:

Data Members:

- Name of the depositor
- Account Number
- Type of account
- Balance amount in the account

Member Functions:

- To assign initial value
- To deposit an amount
- To withdraw an amount after checking

4# Define a class employee having data members as emp_code, dept_code, age, basic, DA, HRA and three member functions as getdata(), putdata(), calculatesalary() to get, display all the values of data members and calculate the total salary by adding basic, DA, HRA. Write this program for 10(ten) employees using an array of objects.

5# Modify the program 2 for handling 10 customers.

WEEK #12

OBJECTIVES

- ❒ To help the students in understanding the concept of Constructor.
- ❒ To help the students in understanding the need for constructors and destructors in their programs.
- ❒ To help the students in understanding the use of copy constructor and dynamic constructors.

OUTCOMES

After completing this, the students would be able to:

- ❒ Write, debug and run constructors and destructors based program in C++.
- ❒ Understand the use of copy constructor and dynamic constructor.
- ❒ Create a simple real life applications based on the constructors and destructors with static members.

PROBLEMS

1# Write a program in C++ to demonstrate default constructor. Create a class having two data members in the private section. Define a default constructor to initialize these data members to initial value and display these values with the help of member function.

2# Write a program in C++ to demonstrate parameterized/constructor overloading constructor. Create a class calculator that contains four data members in it. Initialize data members with different values using parameterized constructor and perform various arithmetic operation over these values and display result on to the computer screen.

3# Create a class called Triangle that stores the length of the base and height of a right triangle in two private instance variables. Include a constructor that sets these values. Define two functions. The first is hypo( ), which returns the length of the hypotenuse. The second is area ( ), which returns the area of the triangle.

4# Create a class for counting the number of objects created and destroyed within various block using constructor and destructors.

WEEK #13

OBJECTIVES

- ❒ To help the students in learning operator overloading and the rules for overloading operators in C++.
- ❒ To help the students in understanding the use of overloading of new and delete operators.

OUTCOMES

After completing this, the students would be able to:

- ❒ Use of binary operator overloading and understand the use of non-member function in overloading.
- ❒ Create a simple real life applications based on Type conversion i.e. type conversion - basic type to class type, class type to basic type, class type to another class type.

PROBLEMS

1# Declare a class Number that contains two data member value1 and value2 of the type of integer, define constructor to give initial value, and perform addition ,subtraction, multiplication and division of these two numbers using operating overloading of +,-,*,/ operator respectively [hint- binary operator overloading using member function]

2# Declare a class Number1 that contains two data member value1 and value2 of the type of integer, define constructor to give initial value, and perform addition, subtraction, multiplication and division of these two numbers using operating overloading of +,-,*,/ operator respectively [hint- binary operator overloading using friend function]

3# Declare a class Number3 that contains a data member value of the type of integer, define constructor to give initial value, and perform unary minus ,increment and decrement this number using operating overloading of -,++,-- operator respectively [hint- Unary operator overloading using member function]

4# Write a program to demonstrate explicit type conversion

- from basic type to user defined data type.
- from User Defined data type to Basic data type data type.

5# Write a program to demonstrate explicit type conversion from one user defined data type to another user defined data type.

6# Define a class complex that contains two data member to store real and imaginary part of the complex number. Create a function to get values from the keyboard into these complex numbers, overload binary + and – to calculate addition and subtraction of two complex numbers respectively using member function.

WEEK #14

OBJECTIVES

- ❒ To help the students in leaning the concept of Inheritance.
- ❒ To help the students in learning how to define a derived classes, ambiguity in multiple and multi-path inheritance in C++.
- ❒ To help the students in understanding the use of different forms of inheritance in their programs.
- ❒ To help the students in understanding the use of multi-path inheritance.

OUTCOMES

After completing this, the students would be able to:

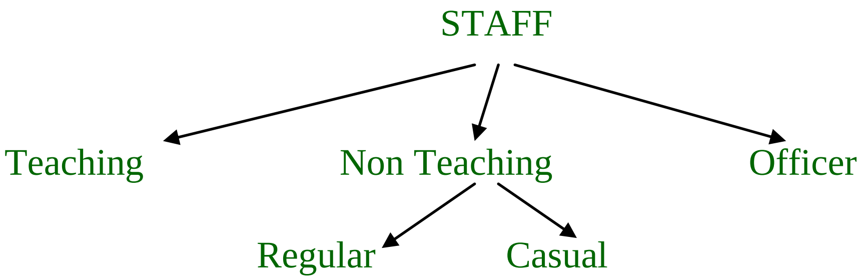
- ❒ Know the working/avoid of ambiguity in multiple and multi-path inheritance.
- ❒ Create a simple real life applications based on virtual base class and overriding member functions.

PROBLEMS

1# Create a class A with some private data members and some public member function, now create a derived class B, that inherits A and having some data members and member functions it's own, in main( ) function access attributes of base class with the help of derived class object to show inheritance concepts.

2# Create a class publication which has title of book and writers name. Create other class sales which account no. of sales for every month (upto 3 months) and then calculate total sales.

3# Write a program to demonstrate the following:



4# Write a program to solve the ambiguity problem in inheritance where two different classes are inherited from single base class and a new class is derived from these two derived classes. How this problem is solved with the help of virtual base class concept.

WEEK #15

OBJECTIVES

- ❒ To help the students in learning the concept of binding in C++.
- ❒ To help the students how to write a sample program based on early binding and late binding in C++.
- ❒ To help the students in understanding the use of binding, virtual functions in their programs.
- ❒ To help the students in understanding the use of polymorphism.
- ❒ To help the students in learning the concept of streams in C++.
- ❒ To help the students in understanding the use of hierarchy of file stream classes in their programs.
- ❒ To help the students in understanding the use of Exception handling mechanism in their programs.
- ❒ To help the students in understanding the use of throwing mechanism and catching mechanism.

OUTCOMES

After completing this, the students would be able to:

- ❒ Write, debug and run a sample program in C++.
- ❒ Differentiate between virtual functions and pure virtual functions.
- ❒ Create a simple real life applications based on the polymorphism.
- ❒ Differentiate the way of reading/writing of files, accessing records randomly, updating files.
- ❒ Create a simple real life applications based on the file handling.
- ❒ Understand hierarchical error handling: **specific errors** (think: **derived classes**) can be handled closer to where the error occurred; for instance deep in the internals of some library. More general errors (think: base classes) can

be handled higher up in the hierarchy; for instance in client code that uses the library in C++.

- ❒ Differentiate try-catch blocks and throw.
- ❒ Create a simple real life applications based on the exceptional Handling statements.

PROBLEMS

1# Write a program to use ‘this’ pointer to find elder from two person. Define a class Person to store age of the person. Define constructor/member function to give initial value to the data member age. And then define a function elder to compare ages of two different persons using this pointer to find out the elder person.

2# Create a simple “shape” hierarchy: a base class called Shape and derived classes called Circle, Square, and Triangle. In the base class, make a virtual function called draw ( ), and override this in the derived classes. Make an array of pointers to Shape objects that you create on the heap (and thus perform up casting of the pointers), and call draw ( ) through the base-class pointers, to verify the behavior of the virtual function. If your debugger supports it, single-step through the code.

3# Write a small program to show the difference between calling a virtual function inside a normal member function and calling a virtual function inside a constructor. The program should prove that the two calls produce different results.

4# Write a program in C++ to calculate mean value of n numbers using friend function.

5# Write a program to accept five different numbers by creating a class called friendfunc1 and friendfunc2 taking 2 and 3 arguments respectively and calculate the average of these numbers by passing object of the class to friend function.

6# Write a program in C++ to display student's information using friend function.

7# Write a C++ program to write text in the file. Read the text from the file from end of file. Display the contents of the file in reverse order.

8# Write a C++ program to count the no. of characters present in the file.

9# Create a class with a main( ) that throws an object of class Exception inside a try block. Give the constructor for Exception a String argument. Catch the exception inside a catch clause and print the String argument. Add a finally clause and print a message to prove you were there.

10# Create your own exception class using the 'extends' keyword. Write a constructor for this class that takes a String argument and stores it inside the object with a String reference. Write a method that prints out the stored String. Create a try-catch clause to exercise your new exception.