3241

Paper IV: Practical

Note: Students are required to perform all the experiments. In the examination two exercises selecting one from each part will be set.

Marks Distribution:

Part A: 24

Program coding: 12

Program execution, result and documentation: 12

Part B: 24

Algorithm and flow chart: 8

Program coding: 8

Program execution, result and documentation: 8

Viva: 12

Record: 15

Part A

Object Oriented Programming (using C++) Lab

1. (a) Write a function power() to raise a number m to a power n. The function takes a double value for m and int value for n, and returns the result correctly. Use a default value of 2 for n to make the function to calculate squares when this argument is omitted. Write a main that gets the value of m and n from user to test the function.
   (b) Write a function that performs the same operation as above but takes an int value for m. Both the functions should have the same name. Write a main that calls both the functions. Use concept of overloading.
2. Define a class 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 values, to deposit an amount, to withdraw an amount after checking the balance. Write a main program to test the program.
3. Write a class to represent a vector (a series of float values). Include member functions to perform the following tasks: To create the vector, to modify the value of a given element, to multiply by a scalar value, to display the vector in the form (10, 20, 30, ...). Write a program to test your

class.

4. A bookshop maintains the inventory of books that are being sold at the shop. The list includes details such as author, title, price, publisher and stock position. Whenever a customer wants a book, the sales person inputs the title and author and the system searches the list and displays whether it is available or not. If it is not, an appropriate message is displayed. If it is, then system displays the book details and requests for the number of copies required. If the requested copies are available, the total cost of the requested copies is displayed, otherwise the message "Required copies not in stock" is displayed. Design a system using a class called books with suitable member functions and constructors. Use new operator in constructors to allocate memory space required.

5. Create a class FLOAT that contains one float data member. Overload all the four arithmetic operators so that they operate on the objects of FLOAT.

6. Create a class matrix and write member functions to implement various matrix operations.

7. Create a class complex for complex numbers and write member functions to implement various operations between complex numbers.

8. Create a class rational for rational numbers and write member functions to implement various operations between rational numbers.

9. Create a class stack and write member functions to implement various stack operations.

10. Create a class queue and write member functions to implement various queue operations.

11. Create a class list and write member functions to implement various list operations.

12. Assume that a bank maintains two kinds of accounts for customers, one called as savings account and the other as current account. The savings account provides compound interest and withdrawal facilities but no cheque book facility. The current account provides cheque book facility but no interest. Current account holders should also maintain a minimum balance and if the balance falls below this level, a service charge is imposed.

   Create a class account that stores customer name, account number and type of account. From this derive the classes cur_acct and sav_acct to make them more specific to their requirements. Include necessary member functions in order to achieve the following tasks: (a) Accept deposit from a customer and update the balance, (b) display the balance, (c) compute and deposit interest, (d) permit withdrawal and update the balance and (e) check the minimum balance, impose penalty, if necessary and update the balance. Do not use any constructors. Use member functions to initialize the class members.

13. Create a base class called shape. Use this class to stroke two double type values that could be used to compute the area of figures. Derive two specific classes triange and rectangle from the base shape. Add to the base class, a member function get_data() to initialize base class data members and another member function display_area() to compute and display the area of figures. Make display_area() as a virtual function and redefine this function in the derived classes to suit their requirements.

Using these three classes, design a program that will accept dimensions of a triangle or rectangle interactively and display the area.

Part B:Linux and Network Lab

Write shell scripts for the following.

1. Accept a pattern and a filename. Check whether the pattern is existing in the file. If found, display "The pattern <pattern> exists in the file <filename>". If it is not found, display the error message as "The pattern <pattern> is not found in the file <filename>".
2. To display the following information.
   (a) Your login diretory is <login directory name>,
   (b) You will get the primary prompt as <primary prompt>,
   (c) Now the date and time is <date> and
   (d) You can execute files in the directories <path>.
3. Accept the name of a file, subdirectory name (which this shell script has to create) and the directory under which this specified subdirectory has to be created. Create the new subdirectory at the directory to this newly created directory.
4. Accept the following information and storing it in a file: Cassette No., movie name, language, price and date of release. The user should be given facility to enter as many records as he wants.
5. The first parameter has to be a directory name. It should be followed by filenames. There can be more than 10 filenames. These specified files should be copied to the directory name passed as the first parameter. The script should also check whether the directory is already existing. If not, it has to be created. The program should also check for the existence of the individual files, whether they are existing and readable.
6. To delete the specified record from the library file. It should accept a book number, check whether it is existing. If so, display the current details, ask for confirmation and then delete it from the file.
7. Accept a word and display its length.
8. Accepting a word from the user and display the reverse of the word.
9. Accept a filename. Assign a read and write permission to others for that file, only if it is not having these permissions at present.
10. Temperature conversion from centigrade to Fahrenheit and vice versa.
11. Accept a number n and a word. It then prints the word n times, one word per line or just on one line (the option should be available).
12. To print the usual date output as default and has options for printing just the time, hust the day-month-year or just the day of the week.
13. To find out whether the given file is a block, character or executable file.
14. Accepts a login name and reports when person logs in. It also sends a greeting to the person.
15. To change your prompt to your login name.

Networking Lab

1. Configure peer to peer Network under Linux
2. Configure Switched Ethernet LAN
3. Configure DHCP
4. Configure File Server