

Roll No. :

--	--	--	--	--	--	--	--	--	--

ANNA UNIVERSITY :: CHENNAI
B.E. / B.Tech. (Full Time) DEGREE END SEMESTER EXAMINATIONS, NOV / DEC 2011
ELECTRONICS & COMMUNICATION ENGINEERING BRANCH
THIRD SEMESTER – (REGULATIONS 2008)
CS9211 - DATA STRUCTURES AND OBJECT ORIENTED PROGRAMMING IN C++

Time : 3 Hrs.

Max. Marks : 100

Answer ALL Questions

PART – A (10 X 2 = 20 Marks)

1. List atleast four new operators added by C++ to C which aid OOP.
2. How does an inline function differ from a preprocessor macro?
3. When do we declare a member of class 'static'? Give example.
4. What is a virtual function? Why do we need virtual functions?
5. Define ADT. Give any two examples.
6. Give the function to delete an item from a dynamic queue.
7. Draw the expression tree for the expression : $(a + b * c) + (d + e) + f$
8. What is meant by Rehashing?
9. Give the different ways of representing a Graph with example.
10. Brief with an example the algorithm design technique, Dynamic Programming.

PART – B (5 X 16 = 80 Marks)

- 11.i) Assume that a bank maintains two kinds of accounts for customers: the savings account and the 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 balance falls below this level, a service charge is imposed. Create a class *Account* that stores *customer name*, *account number* and *account type*. From this derive *Cur-acct* and *Sav-acct* to make them more specific to their requirements. Include necessary member functions in order to achieve the following tasks: [12]
- a) Accept deposit from a customer and update balance.

- b) Display the balance.
- c) Compute and deposit interest.
- d) Permit withdrawal and update the balance.
- e) Check for minimum balance, impose penalty if necessary and update the balance.

11. ii) Illustrate the concept of virtual functions with an example. [4]

12. a) Write an application in C++ for matrix addition and multiplication using the following C++ features: [16]

- (i) Use constructors for initializing the matrix objects.
- (ii) Use constructors for initializing the resultant matrix to zero.
- (iii) Use operator overloading member function for matrix addition.
- (iv) Use friend function for matrix multiplication.
- (v) Find the sum of the elements of the product matrix using class to basic data type type-conversion.

Or

12. b) i) Illustrate with suitable example the different kinds of class type conversions. [8]

b) ii) Write a C++ program to copy the contents of an object of class flowers (name, family, color, number of petals, place of origin) to another object of same class. Also define an operator overloaded function to return the flower object that has more petals. [8]

13. a) i) Write an array implementation of self-adjusting lists. A self-adjusting list is like a regular list, except that all insertions are performed at the front, and when an element is accessed by a 'search', it is moved to the front of the list without changing the relative order of the other items. [10]

a) ii) Discuss the advantages of linked lists over arrayed lists with suitable examples. [6]

Or

13. b) i) Give procedures to perform the stack Push and Pop operations and a third operation Find-min, which returns the smallest element in the data structure. [8]

b) ii) Given input { 4371,1323,6173,4199,4344,9679,1989} and a hash function $h(X) = X \pmod{10}$, show the resulting: [8]

1. Separate chaining hash table.
2. Open addressing hash table using linear probing.
3. Open addressing hash table using quadratic probing.
4. Open addressing hash table with second hash function $h_2(X) = 7 - (X \pmod{7})$

14. a) i) Write functions to perform the following operations on a BST: [5+3]

1. Insertion
2. Findmin

a) ii) Construct a BST by inserting 3,1,4,6,9,2,5,7 into an initially empty BST. Show the results of splaying the nodes 1, 6 and 2 one after the other of the constructed BST. [8]

Or

14. b) i) Write functions to perform the following operations on a BST: [6+2]

1. Deletion
2. Preorder traversal

b) ii) Show the result of inserting 2, 1, 4, 5, 9, 3, 6, 7 in to an initially empty AVL tree. [8]

15. a) i) Explain how the Dijkstra's Algorithm uses the Greedy algorithm design technique in finding the single source shortest path of the graph of fig. 15 – a. [8]

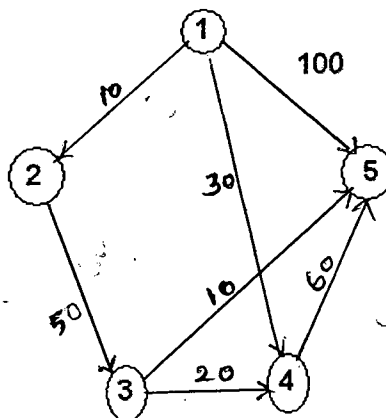


Fig – 15-a

- a) ii) Write a function for shell sort and illustrate using the following list of numbers: 81, 94, 11, 96, 12, 35, 17, 95, 28, 58 [8]

Or

15. b) i) Define divide and conquer algorithm. Explain with an example any sorting algorithm that uses divide and conquer approach. [8]
- b) ii) Explain the Prim's algorithm to find the Minimum Spanning tree for the graph of fig. 15 -b. [8]

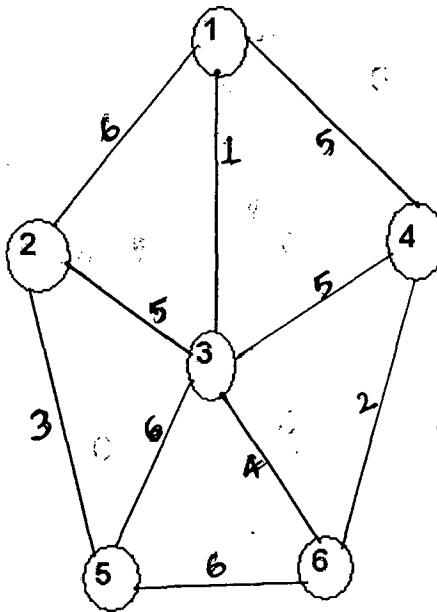


Fig - 15-b