DHARMSINH DESAI UNIVERSITY, NADIAD
FACULTY OF TECHNOLOGY
B.TECH. SEMESTER IV [IT]
SUBJECT: (IT-406) DATA STRUCTURES AND ALGORITHMS

Examination	**: Block Sessional**	**Seat No.**	**: _____**
Date	**: 10/04/2013**	**Day**	**: Wednesday**
Time	**:**	**Max. Marks**	**: 36**

INSTRUCTIONS:
1. Figures to the right indicate maximum marks for that question.
2. The symbols used carry their usual meanings.
3. Assume suitable data, if required & mention them clearly.
4. Draw neat sketches wherever necessary.

Q.1 Do as directed.

(a) The postfix form of the expression (A+ B)*(C*D - E)*F / G is _____ **[2]**

(b) What is the difference between linear and non-linear data structure? **[2]**

(c) Draw a complete binary tree with exactly six nodes. **[2]**

(d) Which sorting algorithm is best if the list is already sorted? Why? **[2]**

(e) If the address of A[1][1] and A[2][1] are 1000 and 1010 respectively and each element occupies 2 bytes then the array has been stored in _____ order. **[2]**

(f) Define the terms: (i) Graph (ii) Hash Function **[2]**

Q.2 Attempt *Any Two* from the following questions. **[12]**

(a) Write an algorithm to construct an Expression Tree from given postfix expression. Construct expression tree for following postfix expression. **[6]**

AB + C * DE - - FG + ^

(b) Create AVL (Height-Balanced) tree for the following sequences with each rotation specified: **[6]**

3,2,1,4,5,6,7,16,15,14,13,12,11,10,8,9

(c) What do you mean by hashing? Explain any four popular hash functions. **[6]**

Q.3 (a) Make a BST for the following sequence of numbers. **[6]**

45, 36, 76, 23, 89, 115, 98, 39, 41, 56, 69, 48

Traverse the tree in Preorder, Inorder and Postorder.

(b) Write an algorithm to create a circular linked list. Write functions to do the following operations: **[6]**

(i) Insert a new node at the end

(ii) Delete the first node

OR

Q.3 (a) What are circular queues? Write down routines for inserting and deleting elements from a circular queue implemented using arrays. **[6]**

(b) Construct a Red-Black tree by inserting the following elements in the order of their occurrence: **[6]**

1, 2, 3, 4, 5, 6