Question No : 1 of 26 Marks: 1 (Budgeted Time 1 Min)

An array x of 100 integers is declared as ,

Answer (Please select your correct option)

- int x = [100] ;
- int x[100] ;
- integer array x = 100 ;
- integer x [] = 100 ;

Question No : 2 of 26 Marks: 1 (Budgeted Time 1 Min)

What is the maximum depth of recursive calls a function may make?

Answer (Please select your correct option)

- 1
- 2
- n (where n is the argument) ✓
- There is no fixed maximum

Question No : 3 of 26 Marks: 1 (Budgeted Time 1 Min)

A Linear Data Structure is the data structure in which data elements are arranged in a sequence or a linear list. Which of the following is Non Linear Data Structure?

Answer (Please select your correct option)

- Arrays
- Linked Lists
- Binary Search Trees ✓
- Stack

Question No : 4 of 26 Marks: 1 (Budgeted Time 1 Min)

The new operation in C++ for dynamically allocating memory returns,

Answer (Please select your correct option)

- Size of the memory it has allocated
- Pointer to the memory it has allocated ✓
- Both size and pointer to the memory it has allocated
- Value of the memory it has allocated

Question No : 5 of 26 Marks: 1 (Budgeted Time 1 Min)

In_____, a programmer uses two pointers in the node, i.e. one to point to next node and the other to point to the previous node.

Answer (Please select your correct option)

- Linked list ✓
- Doubly-linked list
- Array
- Structure

Question No : 6 of 26 Marks: 1 (Budgeted Time 1 Min)

The expression AB+C* is called ?

Answer (Please select your correct option)

- Prefix expression
- Postfix expression ✓
- Infix expression
- Prefix and Infix expression

Question No : 7 of 26 Marks: 1 (Budgeted Time 1 Min)

Consider the function X as under

```
int X (int& Value)
{
return Value;
}
```

Now **a** and **b** are integers in a calling function. Which one of the following is a valid call to the above function X ?

Answer (Please select your correct option)

- a = X (b) ;
- a = X (&b) ;
- a = X (*b) ; ✓
- a = X (&*b) ;

Question No : 8 of 26 Marks: 1 (Budgeted Time 1 Min)

Each node in Binary Search Tree has

Answer (Please select your correct option)

- 1 pointer
- 2 pointers ✓
- 3 pointers
- 4 pointers

Question No : 9 of 26 Marks: 1 (Budgeted Time 1 Min)

Suppose n is the number of nodes in a complete Binary Tree, then maximum steps required for a search operation are

Answer (Please select your correct option)

- $Log_2 (n+1) -1$ ✓
- $Log_2 (n+1)$
- $Log_2 (n) - 1$
- $Log_2 (n)$

Question No : 10 of 26 Marks: 1 (Budgeted Time 1 Min)

The **next** field in the last node of a singly-linked list is set to_____.

Answer (Please select your correct option)

- NAN
- 1
- NULL ✓
- -1

Question No : 11 of 26 Marks: 1 (Budgeted Time 1 Min)

Consider the following function:

```
void test_a(int n)
{
        cout << n << " ";
        if (n>0)
```

Answer (Please select your correct option)

- 4 2 0
- 0 2 4
- 4 2
- 2 4

Question No : 12 of 26 Marks: 1 (Budgeted Time 1 Min)

A queue where the dequeue operation does not depend upon FIFO, is called :

Answer (Please select your correct option)

- enqueue
- simple queue
- stack
- priority queue ✓

Question No : 13 of 26 Marks: 1 (Budgeted Time 1 Min)

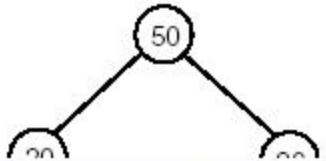
a * (b+c)-d is an example of --------- expression.

Answer (Please select your correct option)

- infix
- prefix
- postfix
- allfix

Question No : 14 of 26 Marks: 1 (Budgeted Time 1 Min)

Below is a Binary Search Tree (BST). If we delete the value 50 from the root node, what would be the value in the root of the remaining tree?

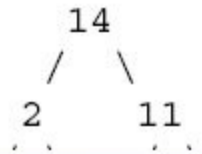


Answer (Please select your correct option)

- 50
- 60
- 70
- 80

Question No : 15 of 26 Marks: 1 (Budgeted Time 1 Min)

Consider the following tree:



Answer (Please select your correct option)

- 2
- 4
- 6
- 9

Question No : 16 of 26 Marks: 1 (Budgeted Time 1 Min)

Which one of the following calling methods does not change the original value of the argument in the calling function?

Answer (Please select your correct option)

- [x] Call by passing the value of the argument ✓
- [ ] Call by passing reference of the argument
- [ ] Call by passing the address of the argument
- [ ] Call by passing pointer of the argument

Question No : 17 of 26 Marks: 1 (Budgeted Time 1 Min)

Which of the following statement is NOT true for reference variable ?

Answer (Please select your correct option)

- Once a reference is created, it cannot be later made to reference another object.
- References cannot be NULL
- References can be uninitialized. ✓
- It is not possible to refer directly to a reference object after it is defined.

Question No : 18 of 26 Marks: 1 (Budgeted Time 1 Min)

Searching an element in an AVL tree takes maximum _____ time (where n is number of nodes in AVL tree)

Answer (Please select your correct option)

- $Log_2(n+1)$
- $Log_2(n+1) - 1$
- 1.44 Log_2n ✓
- 1.66 Log_2n

Question No : 19 of 26 Marks: 1 (Budgeted Time 1 Min)

In the post-order traversal of a binary search tree, nodes process as:

Answer (Please select your correct option)

- Left-subtree , Right-subtree , Root ✓
- Right-subtree , Root , Left-subtree
- Left-subtree , Root , Right-subtree
- Right-subtree , Left-subtree , Root

Question No : 20 of 26 Marks: 1 (Budgeted Time 1 Min)

The main use of AVL tree is:

Answer (Please select your correct option)

- Searching of data ✓
- Storing of data
- Insertion of data
- Security of data

Question No : 21 of 26 Marks: 2 (Budgeted Time 4 Min)

How can the dangling reference problem be avoided ?

Answer (Please click here to Add Answer)

To avoid dangling reference, don't return the reference of a local variable (transient) from a function.

Question No : 22 of 26

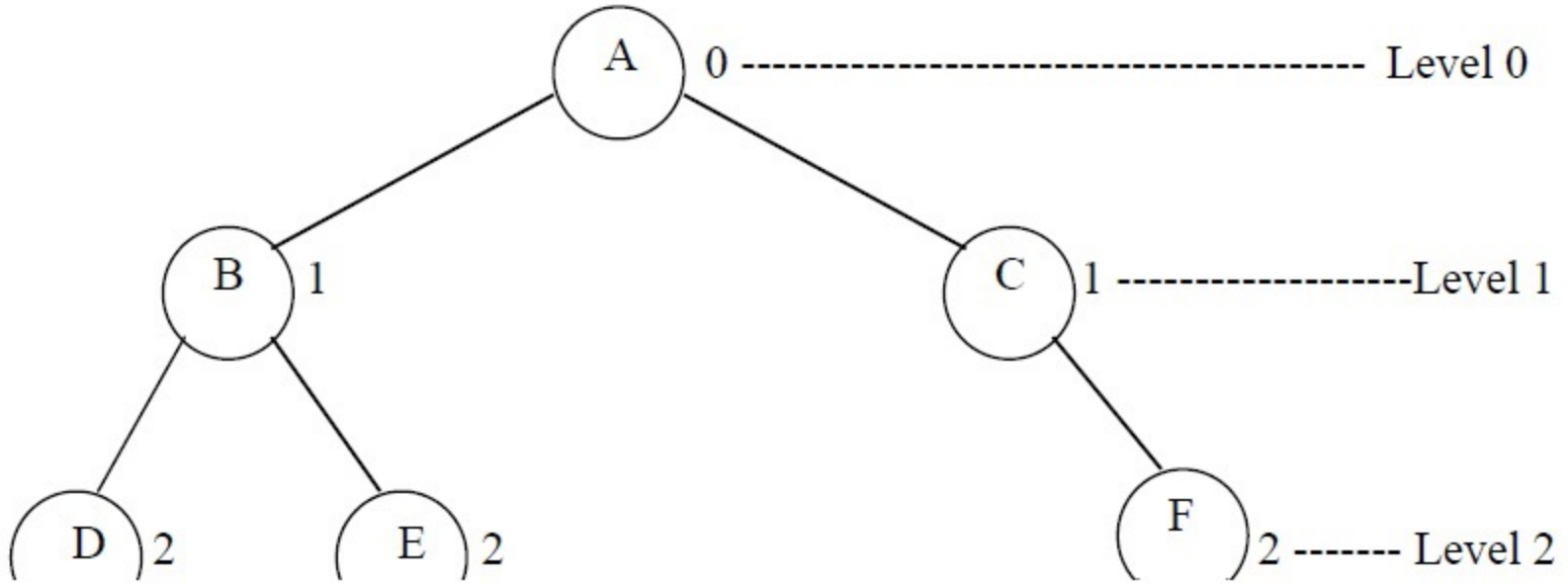
What is "level" in binary tree ?

Answer (Please click here to Add Answer

The level of a node in a binary tree is defined as follows:

- Root has level 0,
- Level of any other node is one more than the level its parent (father).
- The *depth* of a binary tree is the maximum level of any leaf in the tree.

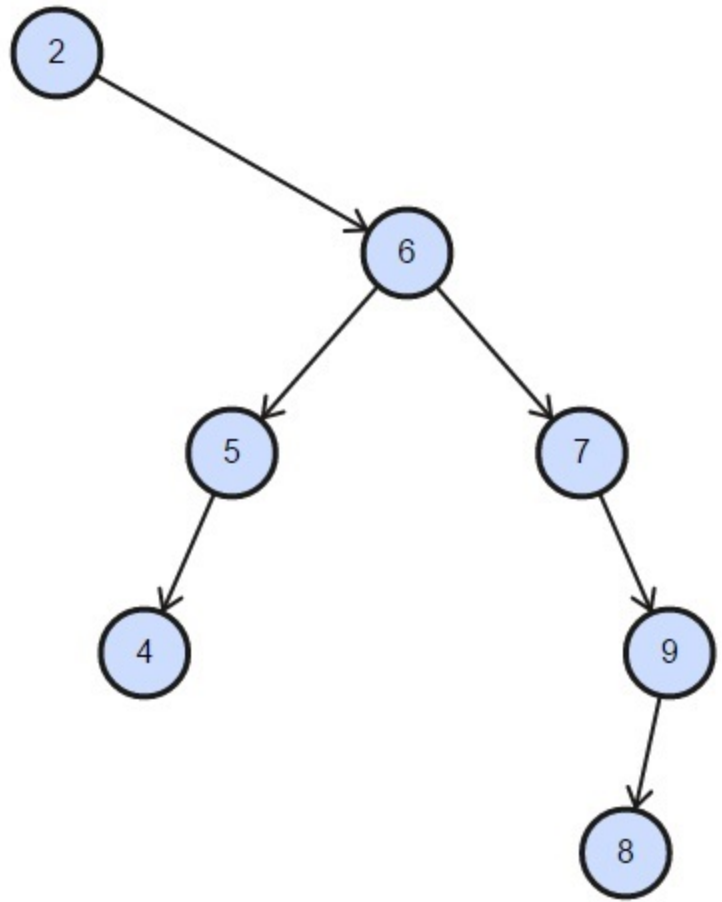
To understand level of a node, consider the following figure 11.13. This figure shows the same tree of figure 11.2, discussed so far.



Question No : 23 of 26

: 3 (Budgeted Time 6 Min)

Draw a BST that is as tall as possible and contains the values '2', '6', '5', '4', '7', '9', '8'.



Answer (Please click here to Add Answer)

Question No : 24 of 26 Marks: 3 (Budgeted Time 6 Min)

Define the following terms

1) Reference variables
2) Dangling reference
3) Const keyword

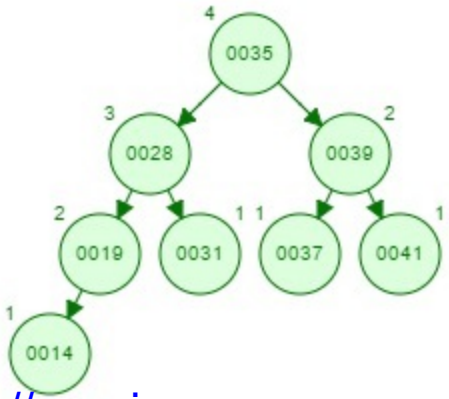
Answer (Please click here to Add Answer)

Ref Vari: A reference variable is an alias, that is, another name for an already existing variable. Once a reference is initialized with a variable, either the variable name or the reference name may be used to refer to the variable.
Dangling: A dangling reference is a reference to an object that no longer exists.
Const: By writing *const*, we are saying that parameter must remain constant for the life of the function.

Question No : 25 of 26 Marks: 5 (Budgeted Time 10 Min)

Show the result of inserting the values **35, 19, 41, 39, 37, 28, 31, 14** into an empty AVL Tree.
You have to show only the complete AVL tree, steps are not required.

Answer (Please click here to Add Answer)



Question No : 26 of 26 Marks: 5 (Budgeted Time 10 Min)

Write down the algorithm (Steps) for level order traversing of a binary tree.

Answer (Please cli

Normal

The name of the method is *levelorder* and it is accepting a pointer of type *TreeNode <int>*. This method will start traversing tree from the node being pointed to by this pointer. The first line (line 3) in this method is creating a queue *q* by using the *Queue* class factory interface. This queue will be containing the *TreeNode<int>* * type of objects. Which means the queue will be containing nodes of the tree and within each node the element is of type *int*.

The line *5* is checking to see, if the *treeNode* pointer passed to the function is pointing to NULL. In case it is NULL, there is no node to traverse and the method will return immediately.

Otherwise at line *6*, the very first node (the node pointed to by the *treeNode* pointer) is added in the queue *q*.

Next is the *while* loop (at line *7*), which runs until the queue *q* does not become empty. As we have recently added one element (at line *6*), so this loop is entered.

At line *9*, *dequeue()* method is called to remove one node from the queue, the element at *front* is taken out. The return value will be a pointer to *TreeNode*. In the next line (line *10*), the *int* value inside this node is taken out and printed. At line *11*, we check to see if the left subtree of the tree node (we've taken out in at line *9*) is present. In case the left subtree exists, it is inserted into the queue in the next statement at line *12*. Next, we see if the right subtree of the node is there, it is inserted into the queue also. Next statement at line *15* closes the *while* loop. The control goes back to the line *7*, where it checks to see if there is some element left in the queue. If it is not empty, the loop is entered again until it becomes empty.