1.What does creating node mean?

1. Defining its structure
2. Allocating memory to it
3. Initialization
4. All of the above

**void** fun1(Node head)

{

**if**(head == **NULL**)

**return**;

fun1(head.next);

System.out.println(head.data);

}

1. Prints all nodes of linked lists
2. Prints all nodes of linked list in reverse order
3. Prints alternate nodes of Linked List

d) Prints alternate nodes in reverse order

2.How many children does a binary tree have?

a) 2  
b) any number of children  
c) 0 or 1 or 2  
d) 0 or 1

3. What is a full binary tree?  
a) Each node has exactly zero or two children  
b) Each node has exactly two children  
c) All the leaves are at the same level  
d) Each node has exactly one or two children

What does creating node mean?

1. Defining its structure
2. Allocating memory to it
3. Initialization
4. All of the above

**void** fun1(Node head)

{

**if**(head == **NULL**)

**return**;

fun1(head.next);

System.out.println(head.data);

}

1. Prints all nodes of linked lists
2. Prints all nodes of linked list in reverse order
3. Prints alternate nodes of Linked List

d) Prints alternate nodes in reverse order

How many children does a binary tree have?

a) 2  
b) any number of children  
c) 0 or 1 or 2  
d) 0 or 1

 What is a full binary tree?  
a) Each node has exactly zero or two children  
b) Each node has exactly two children  
c) All the leaves are at the same level  
d) Each node has exactly one or two children