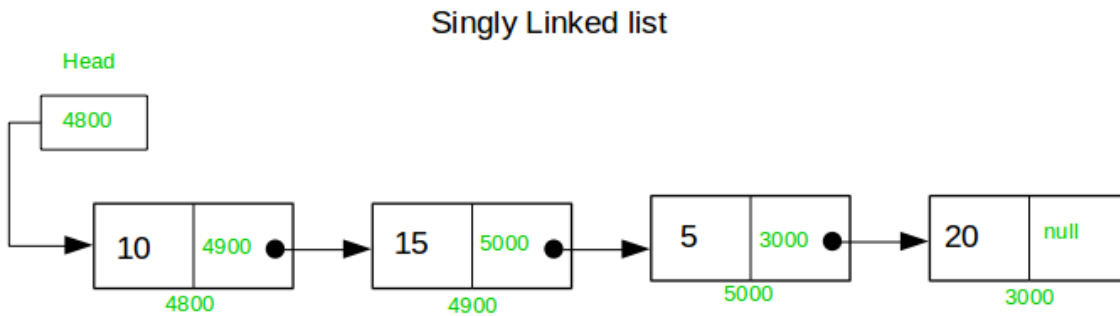
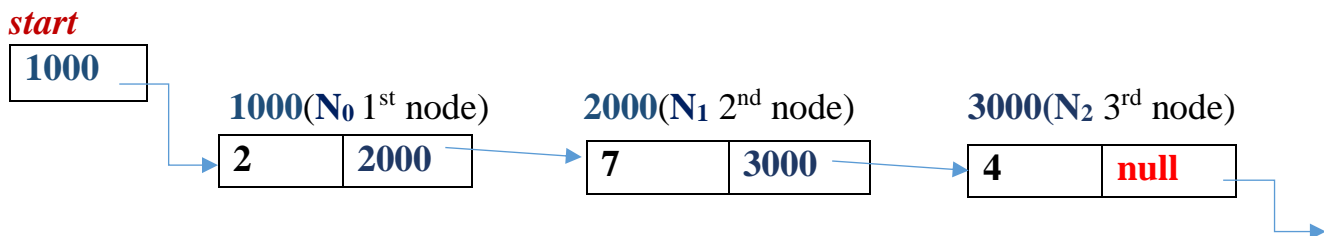


Introduction to circular linked list

In single linear linked list the link part of the last node contains '**null**' value referring to nothing which indicates end or last node of the list.

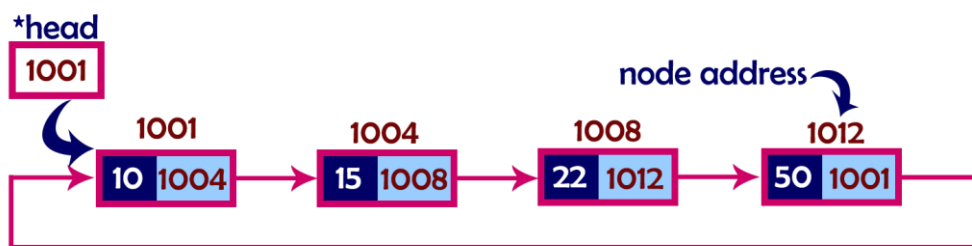


(Fig: 1 single linked list containing an integer value in each node of the list)

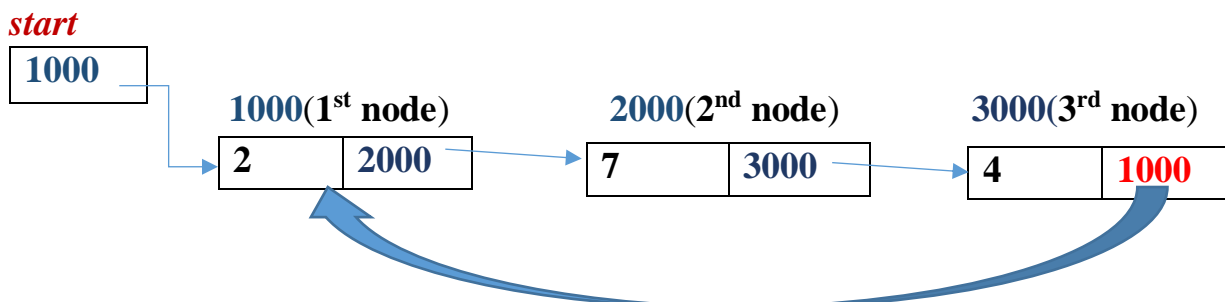


Definition:

It is a linear dynamic data structure consisting of a collection of nodes where the link part of last node refers to the first node of the list forming a circular list.



(Fig: 2 Circular linked list containing an integer value in each node of the list)



Advantage of Circular Single list:

- ➔ We can visit from any node to any other node of the list which is not possible in single linear linked list.

Applications:

- Implementation of queues
- In multi-programming or multi-tasking operating systems where multiple programs are executed in round robin manner.

Operations on single circular linked list

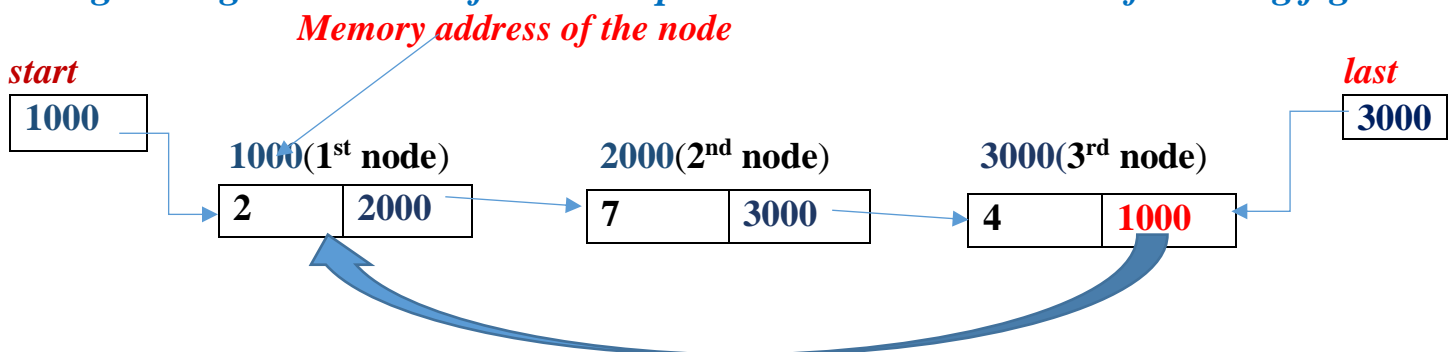
1. Creation
2. Displaying or traversing the list.
3. Insertion of a new node into the list.
4. Deletion of a node from the list.

Implementation of Circular Single Linked lists

- Unlike in single linked list where we use only one special pointer named as 'start' or 'head' that always refers to first nodes, but in circular linked list we use two pointers name as 'start' that refers to first node and 'last' that refers to last nodes address always.
- The last nodes link part refers to first node i.e. it holds first nodes address always.
- The node type class declaration to create nodes is same as single linked list as follows:

```
class NODE          or      class integer_node
{
    int info;
    NODE next;
}
{
    int node_value;
    integer_node nextlink;
}
```

*/*The above class declarations for a circular single linked list is same as single linked list as shown in the above class declarations, where each node will store only a single integer value in information part which is shown in the following figure*/*



Example:

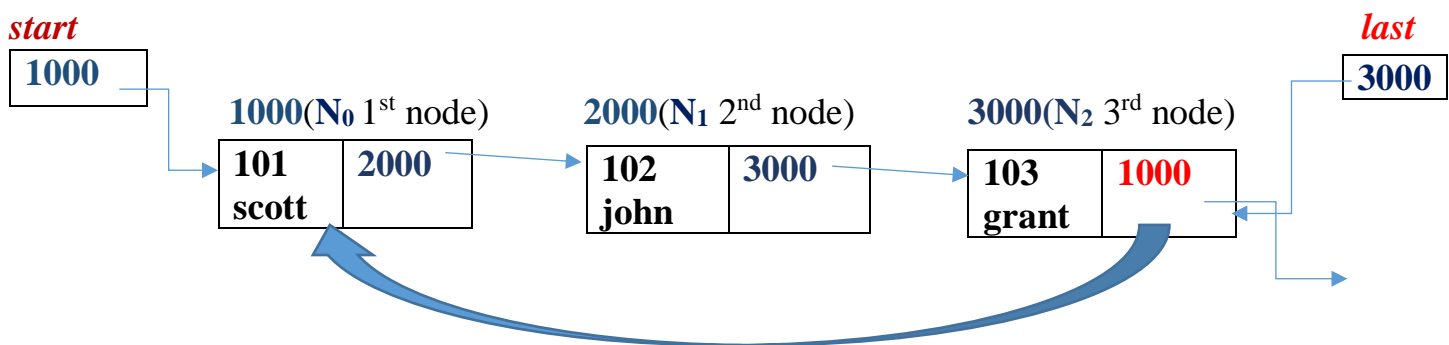
```

class emp_node
{
    int emp_id;
    int emp_name;

    emp_node link;
}

```

/*The above emp_node class can be used to store information of employees of an enterprise where each node of the linked list can store employee id and employee names in information part as shown in the following figure:*/



(Fig: 3 A single linked list containing employee information in each node)