Name : Suryakant Upadhyay

PRN : 20220802043

Batch : A1

1. Write a python code to create a single node in a binary search Tree.

In [1]:
```
class Node:
    def __init__(self,data):
        self.left = None
        self.right = None
        self.data = data
root = Node(10)
root.data
```

Out[1]:
```
10
```

2. Write a python code to insert more than one node in to a binary search tree.

In [2]:
```
class Node:
    def __init__(self, data):
        self.left = None
        self.right = None
        self.data = data


def insert_node(root, data):
    if root is None:
        return Node(data)
    else:
        if data < root.data:
            root.left = insert_node(root.left, data)
        else:
            root.right = insert_node(root.right, data)
    return root


root = None
root = insert_node(root, 5)
root = insert_node(root, 3)
root = insert_node(root, 6)
root = insert_node(root, 1)
root = insert_node(root, 9)


def inorder(node):
    if node is not None:
        inorder(node.left)
        print(node.data)
```

```
        inorder(node.right)


inorder(root)
```

```
1
3
5
6
9
```

3. Write a python code to perform In-order binary search Tree Traversal.

In [3]:
```
class Node:
    def __init__(self, data):
        self.left = None
        self.right = None
        self.data = data

def inorder(node):
    if node is not None:
        inorder(node.left)
        print(node.data)
        inorder(node.right)

root = Node(5)
root.left = Node(3)
root.right = Node(7)
root.left.left = Node(1)
root.right.right = Node(9)
inorder(root)
```

```
1
3
5
7
9
```

4. Write a python code to perform Pre-order binary search Tree Traversal.

In [4]:
```
class Node:
    def __init__(self, data):
        self.left = None
        self.right = None
        self.data = data

def preorder(node):
    if node is not None:
        print(node.data)
        preorder(node.left)
        preorder(node.right)

root = Node(5)
root.left = Node(3)
root.right = Node(7)
root.left.left = Node(1)
root.right.right = Node(9)
preorder(root)
```

```
5
3
1
7
9
```

5. Write a python code to perform Post- order binary search Tree Traversal.

In [5]:
```
class Node:
    def __init__(self, data):
        self.left = None
        self.right = None
        self.data = data

def postorder(node):
    if node is not None:
        preorder(node.left)
        preorder(node.right)
        print(node.data)

root = Node(5)
root.left = Node(3)
root.right = Node(7)
root.left.left = Node(1)
root.right.right = Node(9)
postorder(root)
```

```
3
1
7
9
5
```