LAB#12

Example#1: Write a program to insert data in a binary search tree.

Solution:

```
class Node:
    def __init__(self,left=None,item=None,right=None):
        self.left=left
        self.item=item
        self.right=right

```

```
class BST:
    def __init__(self):
        self.root=None
    def insert(self,data):
        self.root=self.rinsert(self.root,data)
    def rinsert(self,root,data):
        if root is None:
            return Node(None,data,None)
        if data<root.item:
            root.left=self.rinsert(root.left,data)
        elif data>root.item:
            root.right=self.rinsert(root.right,data)
        return root
```

```
t1=BST()
t1.insert(80)
t1.insert(100)
t1.insert(70)
t1.insert(90)
t1.insert(60)
```

Explanation:

```
t1=BST()
```

```
t1=BST()
```

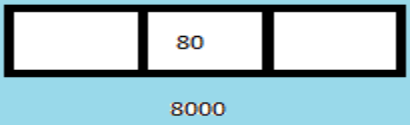
```
def __init__(self):
    self.root=None
```

```
t1.insert(80)
```

```
t1.insert(80)
```

```
def insert(self,data):
    self.root=self.rinsert(self.root,data)
```

```
def rinsert(self,root,data):
    if root is None:
        return Node(None,data,None)
```



self.root=8000

```
t1.insert(100)
```

```
def insert(self,data):
    self.root=self.rinsert(self.root,data)
```

self.root= self.rinsert(8000,100)

```
def rinsert(self,root,data):



    elif data>root.item:
        root.right=self.rinsert(root.right,data)
    return root
```

root.right=None

return root(8000)

```
def rinsert(self,root,data):
    if root is None:
        return Node(None,data,None)
```

1000

return 1000

root.right=1000

Updated table:

root.right=1000

self.root=8000

```
t1.insert(70)
```

```
def insert(self,data):
    self.root=self.rinsert(self.root,data)
```

```
def rinsert(self,root,data):
    if data<root.item:
        root.left=self.rinsert(root.left,data)
    return root
```

data is 70 root is 8000

root.left is None here

return 8000

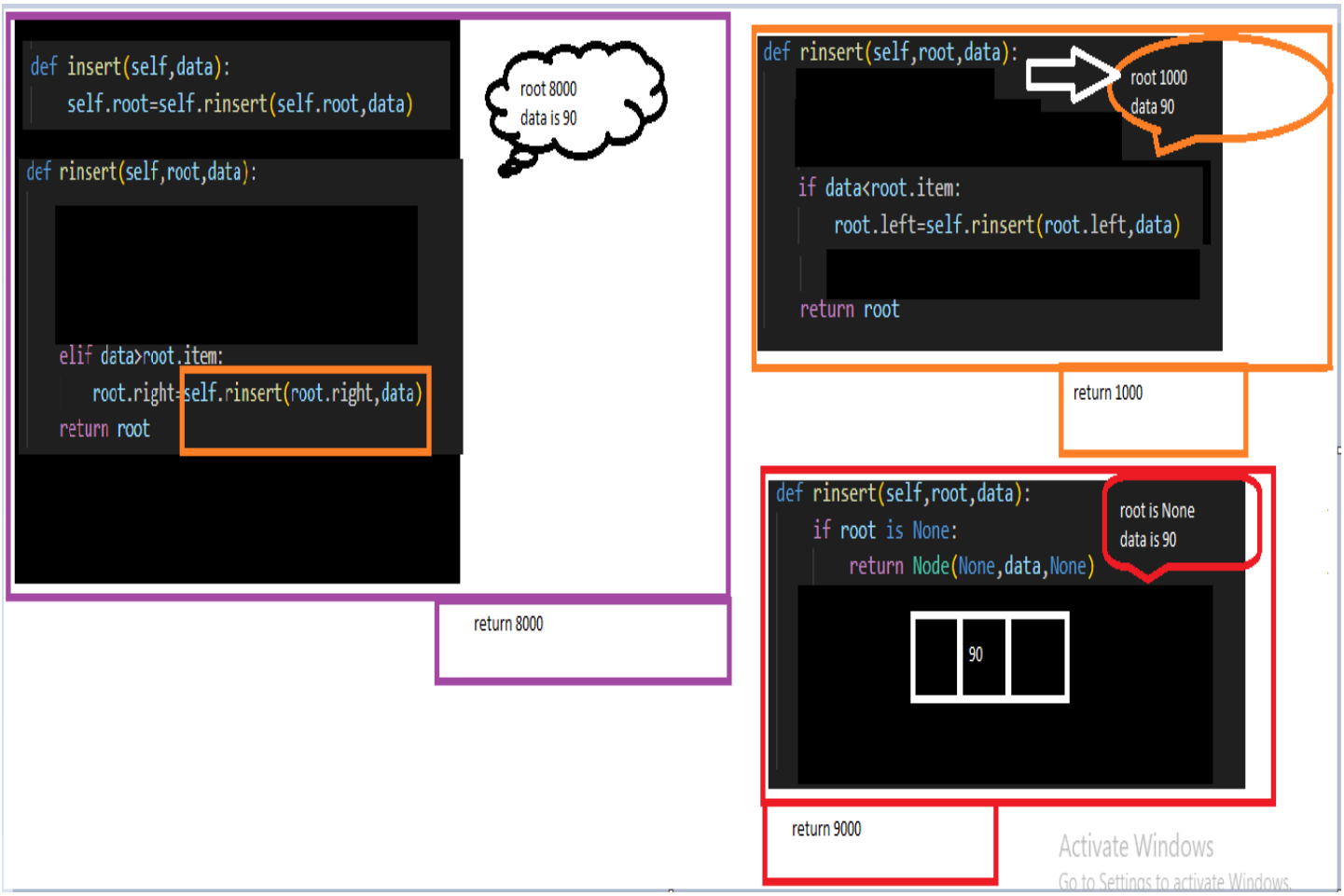
```
def rinsert(self,root,data):
    if root is None:
        return Node(None,data,None)
```

7000

return 7000

self.left=7000

```
t1.insert(90)
```


def insert(self,data):
self.root=self.rinsert(self.root,data)
def rinsert(self,root,data):
elif data>root.item:
root.right=self.rinsert(root.right,data)
return root
root 8000
data is 90
return 8000
def rinsert(self,root,data):
root 1000
data 90
if data<root.item:
root.left=self.rinsert(root.left,data)
return root
return 1000
def rinsert(self,root,data):
if root is None:
return Node(None,data,None)
root is None
data is 90
90
return 9000
Activate Windows
Go to Settings to activate Windows.

```
t1.insert(60)
```

Class Assignment

Q: Modify the program LAB#12 example#1 by creating two trees in this same program. Insert four number of items in the first tree and five number of items in the second tree. Explain the code diagrammatically.

Note:

Implement two binary search trees one by one.

Similar elements are not allowed in a binary search tree.