Create a project called `lab6` (if using Visual Studio). Download the source file called `lab6.cpp` from Blackboard, and use it as the starting point for your assignment.

Your group will complete the implementation of the template class called `BSTNode`, which implements a Binary Search Tree for any base type that defines the < operator and the == operator. The member functions that your group will complete are:

- `void BSTNode<T>::insert(const T& data);`
- `void BSTNode<T>::preOrderDisplay();`
- `void BSTNode<T>::postOrderDisplay();`

Follow the instructions contained in the comments in `lab6.cpp` to get started.

When you're done, the output of supplied driver program (using `BSTNode<T>::inOrderDisplay()` in `operator<<`) should look like this:

```
iroot == 10, 20, 100, 200, 300
sroot == Friday, Monday, Saturday, Sunday, Thursday, Tuesday,
Wednesday
```

When you substitute `preOrderDisplay()` in `operator<<`, the output should look like this:

```
iroot == 100, 10, 20, 200, 300
sroot == Sunday, Monday, Friday, Saturday, Tuesday, Thursday,
Wednesday
```

When you substitute `postOrderDisplay()` in `operator<<`, the output should look like this:

```
iroot == 20, 10, 300, 200, 100
sroot == Friday, Saturday, Monday, Thursday, Wednesday, Tuesday,
Sunday
```

Be prepared to demonstrate the result of any one (or all three) of the above display methods at the lab TA's request (by changing the definition of `operator<<` as needed).

When finished, one member of your group should turn in your completed `lab6.cpp` file on Blackboard.