12. Trees :: Binary Trees :: Java Implementation :: *BinaryTree<E>* Constructors

The *BinaryClass<E>* class has three constructors:

```
// Create a new empty BinaryTree storing no data.
public BinaryTree() {
  this(null);
}

// Create a new BinaryTree with the root storing pData. The left and right subtrees
// are empty.
public BinaryTree(E pData) {
  this(pData, null, null);
}

// Create a new BinaryTree with the root storing pData. Set the left and right
// child subtrees to pLeft and pRight.
public BinaryTree(E pData, BinaryTree<E> pLeft, BinaryTree<E> pRight) {
  Node<E> leftChild  = (pLeft == null)  ? null : pLeft.getRoot();
  Node<E> rightChild = (pRight == null) ? null : pRight.getRoot();
  setRoot(new Node<E>(pData, leftChild, rightChild));
}
```

12. Trees :: Binary Trees :: Java Implementation :: *BinaryTree<E>* Constructors

These would be called in this manner:

```
BinaryTree<Integer> tree0 = new BinaryTree<>();
BinaryTree<Integer> tree1 = new BinaryTree<>(1);
BinaryTree<Integer> tree2 = new BinaryTree<>(2);
BinaryTree<Integer> tree3 = new BinaryTree<>(3, tree1, tree2);
tree3.traverse(BinaryTree.LEVEL_ORDER, this);
```