Task 1: Create a node for a tree and include a constructor (empty) [Hint : A node which consists of 1 data part and 2 refs ( 1 Left ref and another right ref)]

Example :

class TreeNode {

int value;

TreeNode left, right;

TreeNode(int item) {

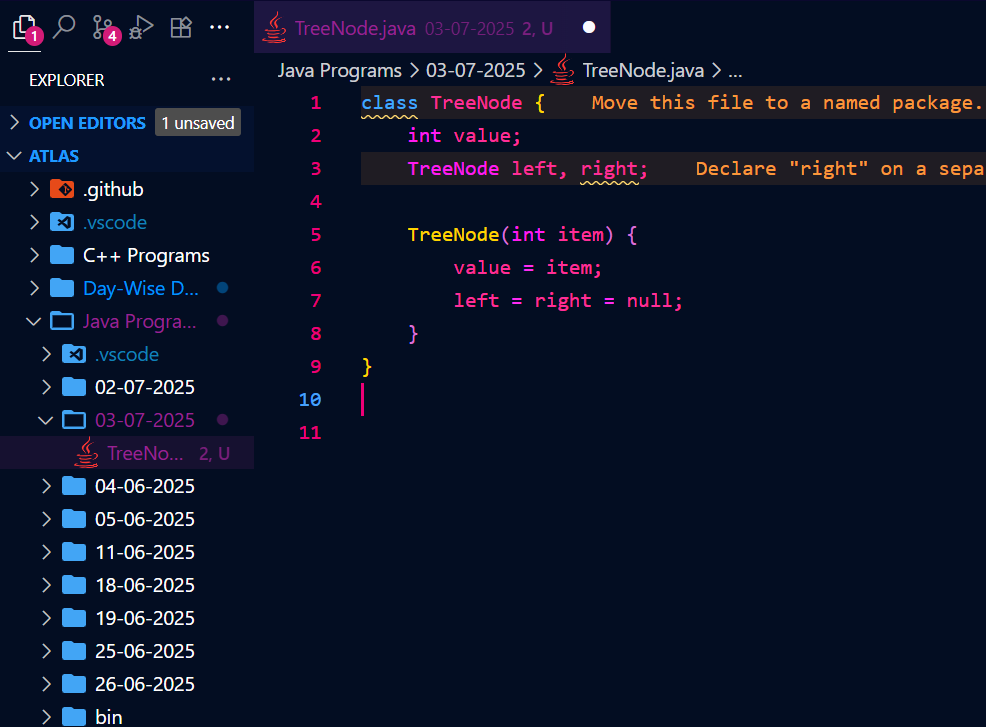
value = item;

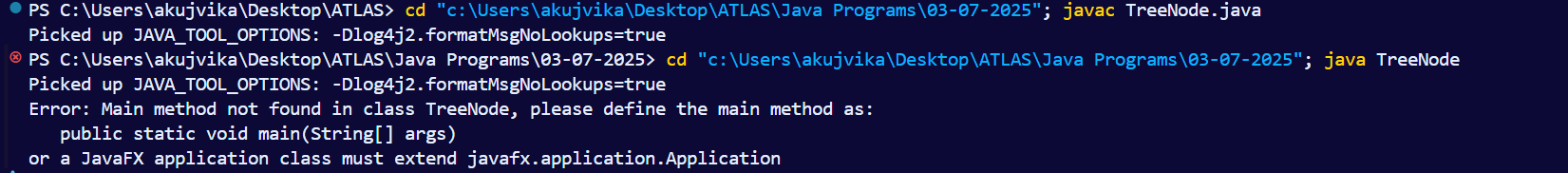
left = right = null;

}

}

Solution : This class represents a node in a binary tree. Each node contains an integer value and pointers to its left and right children.





Task 2: Create a class named Binarty Search tree in which you have 2 insert operations

1 insert —----> for inserting if the tree is empty

1 insert —----> for inserting if the tree is 1 or more nodes

TreeNode insertVal(TreeNode node, int value) {

if (node == null) {

node = new TreeNode(value);

return node;

}

if (value < node.value) {

node.left = insertVal(node.left, value);

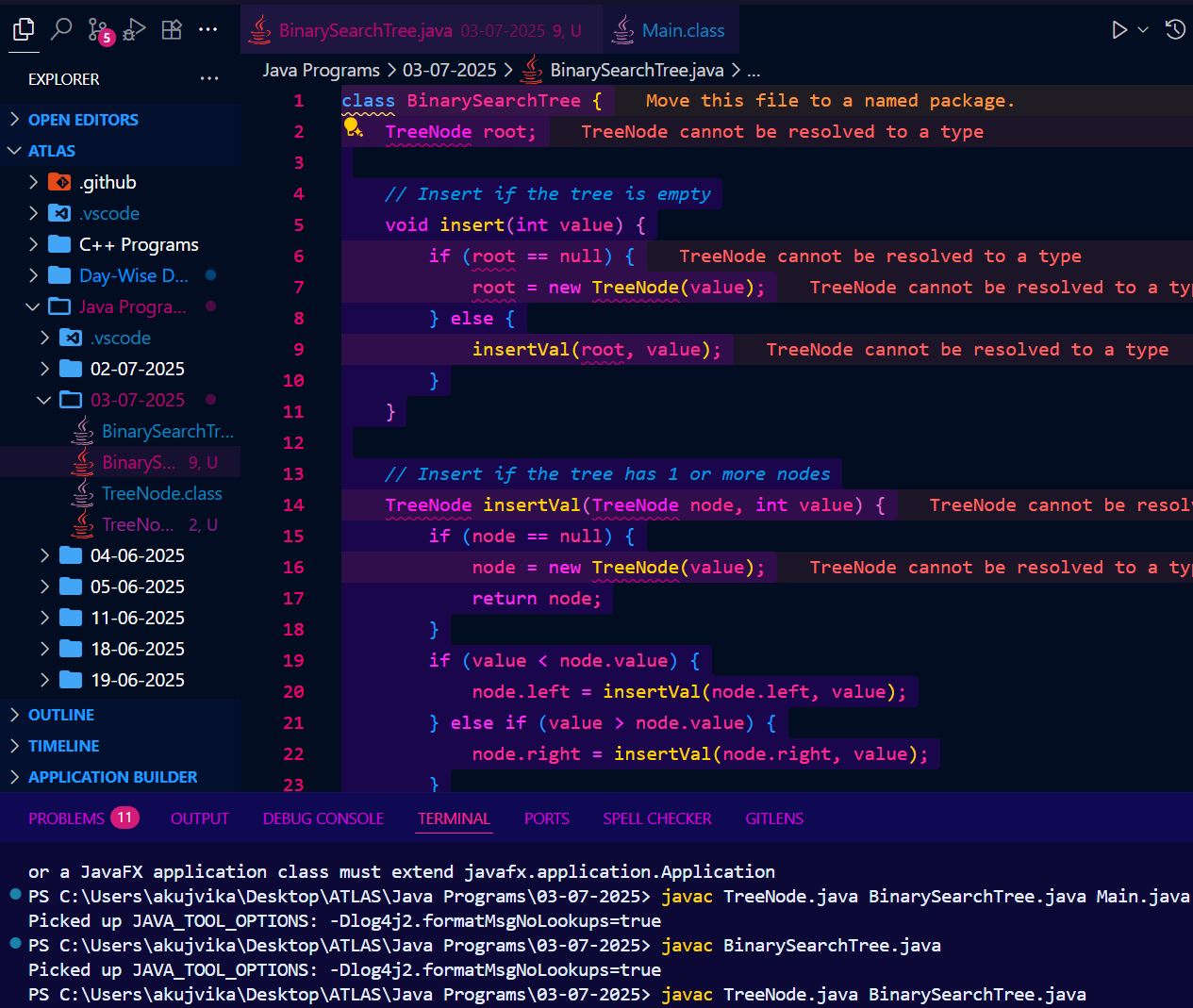
} else if (value > node.value) {

node.right = insertVal(node.right, value);

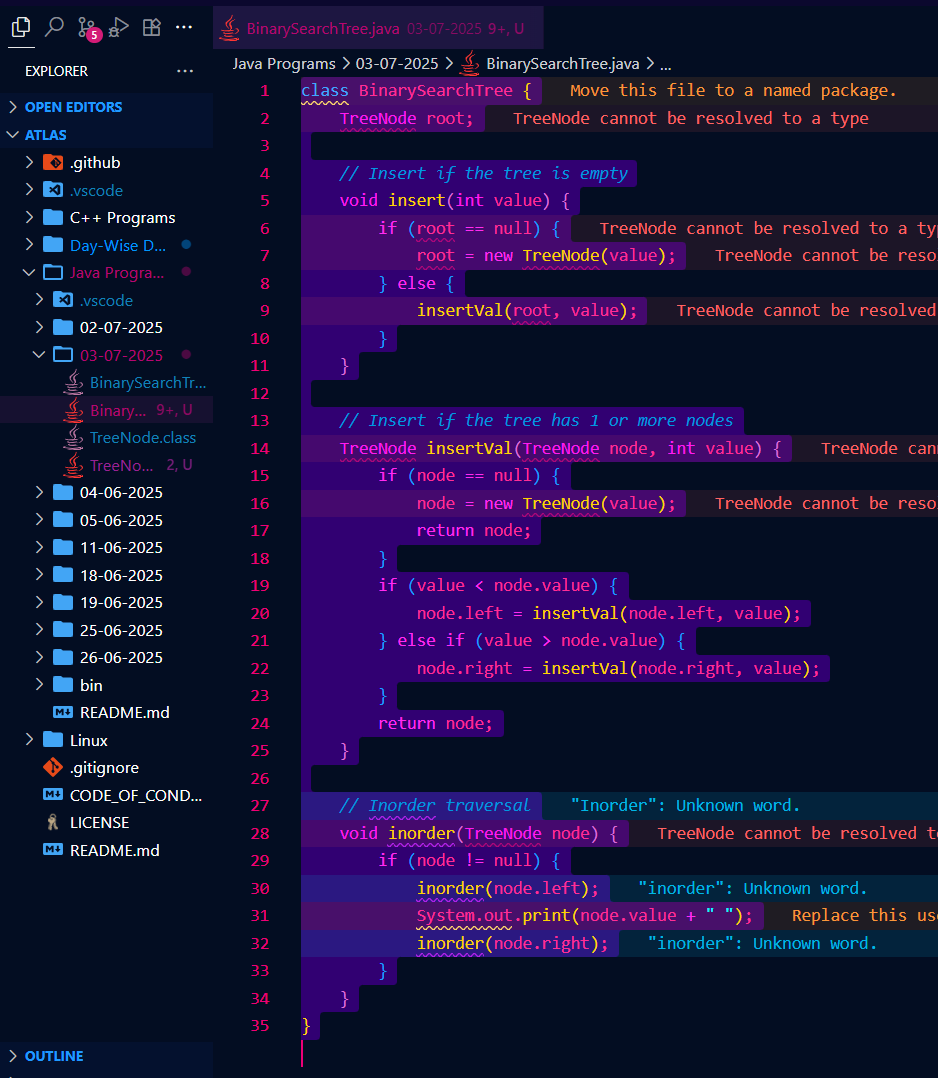
}

return node;

}

Solution : 

Task 3: Ionorder travel of the above code snippets from task 1 and Task 2

Solution : 

Task 4: Create a main method Task 1, 2 and 3 and run the code..

Solution : 