# Lab 6-2 Binary Search Tree Assignment Details

The focus of these problems will be working with information extracted from a municipal government data feed containing bids submitted for auction of property. The data set is provided in two comma-separated files:

1. eBid\_Monthly\_Sales.csv (larger set of 17,937 bids)
2. eBid\_Monthly\_Sales\_Dec\_2016.csv (smaller set of 179 bids)

This assignment is designed to explore the binary search tree algorithm using bids loaded from a CSV file.

We provide a starter console program that uses a menu to enable testing of the hash table logic you will complete. It also allows you to pass in the path to the bids CSV file to be loaded, enabling you to try both files. In this version the following menu is presented when the program is run:

**Menu:**

**1. Load Bids**

**2. Display All Bids**

**3. Find Bid**

**4. Remove Bid**

**9. Exit**

**Enter choice:**

The BinarySearchTree.cpp program is partially completed - it contains empty methods representing the programming interface used to interact with a hash table. You will need to add logic to the methods to implement the necessary behavior. Here is the public API for BinarySearchTree that you have to complete:

public:

BinarySearchTree();

virtual ~BinarySearchTree();

void Insert(Bid bid);

void Remove(string bidId);

Bid Search(string bidId);

You will need to perform the following steps to complete this activity:

Setup: Begin by creating a new C++ Project with a Project Type of "Hello World C++ Project"

1. Name the project ‘BinarySearchTree’, remember to pick the correct compiler in **Toolchains** and click Finish. This will create a simple BinarySearchTree.cpp source file under the **/src** directory.
2. Download the starter program files and copy them to the project’s **/src** directory, replacing the existing auto-generated one. Remember to right-click on the project in the Project Explorer pane on the left and **'Refresh'** the project so it adds all the new files to the src folder underneath.
3. Because this activity uses C++ 11 features you must follow the instructions under “C++ Compiler Version” in the *C++ Development Installation guide* to add **-std=c++11** compiler switch to the Miscellaneous settings.

Task 1: Define structures for tree node, housekeeping variables

Task 2: Implement inserting a bid into the tree

Task 3: Implement removing a bid from the tree

Task 4: Implement searching the tree for a bid

Hint: Lab3-3 used a Node structure for implementing a linked list.

Here is sample output from running the completed program:

> ./BinarySearchTree ~/Downloads/eBid\_Monthly\_Sales\_Dec\_2016.csv

> BinarySearchTree.exe Downloads\eBid\_Monthly\_Sales\_Dec\_2016.csv

## Load bids from CSV and display the hash table contents:

| **Example Input** | **Choice: 1** | **Choice: 3** |
| --- | --- | --- |
| **Display** | Menu:  1. Load Bids  2. Display All Bids  3. Find Bid  4. Remove Bid  9. Exit  Enter choice: 1 | Menu:  1. Load Bids  2. Display All Bids  3. Find Bid  4. Remove Bid  9. Exit  Enter choice: 3 |
| **Output** | Loading CSV file eBid\_Monthly\_Sales.csv  179 bids read  time: 6795773 clock ticks  time: 6.79577 seconds | 98109: Whirlpool Washer & Dryer | 225.46 | Enterprise  time: 183 clock ticks  time: 0.000183 seconds |

Note that it took only 183 clock ticks to search nearly 18,000 records in a binary tree.