**BINARY TREE**

**Add**

**Parameters – Node newNode**

**Return – Node addedNode**

Search through the tree for the right spot to add newNode

Add pointer to newNode from its parent

If newNode has a child, save its pointer and add it to newNode

**Return –** Node that was added (for log file output)

**Delete**

**Parameters – Node deleteNode (has same name as node to be deleted)**

**Return – Node deletedNode**

Search tree for node with same name as deleteNode

If it exists, delete it

Adjust its parent and children pointers accordingly

**Return –** copy of Node that was deleted (for log file output)

**Search**

**Parameters – Node searchNode (has same name as node searching for)**

**Return – Node searchedNode**

Search tree for node with same name as searchNode

**Return –** Node that was searched, null if it doesn’t exist

**MAIN**

**Edit**

**Parameters – String name, Int fieldNum, Int value, PrintWriter output**

**Return – nothing**

Search BinTree for node with same name

Edit either the high score or number of plays (based on fieldNum)

Print output to log file

**Edit (overloaded)**

**Parameters – String name, Int fieldNum, String value, PrintWriter output**

**Return – nothing**

Search BinTree for node with same name

Edit the initials (since value is a string in this overloaded function)

Print output to log file

**PrintDatabase**

**Parameters – PrintWriter output**

**Return – nothing**

Traverse the entire BinTree

For each node Print <name>, <high\_score>, <initials>, <plays>, <$><revenue><\n>

**Main**

Open input files

Open output files

If batch file opens properly

While not EOF

Read line, determine what action to take

Perform action and print output to log file

Write entire BinTree to database file

Close files