Name: Islam Osama Nwishy

ID#: 900170200

Assignment 5 – Design Document

* **BinaryTree [Class]:**
  + **KeyType:** Template class that holds the type of the key of a tree node
  + **DataType:** Template class that holds the type of the data inside the tree
  + **Public:**
    - **BinaryTree(constructor):** Sets the Root to NULL.
    - **~BinaryTree(Destructor):** Calls a Function to delete the tree and then it deletes the root
    - **Insert():** Takes a key and data and inserts it to the tree in its appropriate place
    - **Empty():** returns true if root equals null
    - **Search():** searches for an element of a given key
    - **Retrieve():** Returns the data of an element of a given key
    - **Traverse():** Traverses the tree in-order and outputs its values to the screen
    - **Traverese(ofstream):** Traverses the tree in level-order and outputs its values to a file
    - **Preorder():** Traverses the tree in Pre-order and outputs its values to the screen
    - **LevelOrder():** Traverses the tree in Level-order and outputs its values to the screen
    - **Remove():** removes an element of a given key
    - **Update():** updates an element of a given key
    - **DeleteTree():** Deletes the whole tree
    - **ReverseRetrive():** outputs the keys of the elements where its data are more than or equal a certain threshold
  + **Private:**
    - **TreeNode[class]:**
      * **Public**
        + **Key:** keytype variable that holds the key of a node
        + **Data:** data type variable that holds the data of a node
        + **Right:** a pointer to a right treenode
        + **Left:** a pointer to a left treenode
    - **Insert2():** Takes a key and data then recursively finds its appropriate place in the tree using the root and inserts it
    - **Search2():** recursively searches for an element of a given key and the root
    - **Retrieve2():** Returns the data of an element of a given key after searching for it recursively
    - **Traverse2():** Traverses the tree recursively in-order given the root and outputs its values to the screen
    - **Traverese2(ofstream):** Traverses the tree recursively in level-order given the root and outputs its values to a file
    - **Preorder2():** Traverses the tree recursively in Pre-order given the root and outputs its values to the screen
    - **LevelOrder2():** Traverses the tree recursively in Level-order given the root and outputs its values to the screen
    - **Remove2():** removes an element of a given key after finding it recursively given a root
    - **Update2():** updates an element of a given key after finding it recursively given a root
    - **DeleteTree2():** Deletes the whole tree by traversing it recursively and deleting each node
    - **ReverseRetrive2():** outputs the keys of the elements where its data are more than or equal a certain threshold by traversing the tree recursively and checking each element
    - **PrintLevel():** outputs the data of a given level in the tree to the screen
    - **PrintLevel2():** outputs the data of a given level in the tree to a file
    - **Findparent():** finds the parent of a child where the key of the child is equal to a given key
    - **FindMin():** finds the minimum node in a tree starting from a given parent