

ASSIGNMENT 1 REPORT

The object-oriented design I opted for was a binary search tree due the time complexity of the code as a large input size was being handled from the knowledge base. An array version of the same code was also made and the difference in efficiencies can clearly be noticed when loading the knowledge base.

CLASSES USED:

- BSTNode – This class served the purpose of holding all the data required for a single node as well as a left and right pointer that would allow it to be linked to nodes that were added in future.
- BinarySearchTree – This class served the purpose of defining the structure of the binary search tree as well as the methods such as inserting and sorting the nodes once they had been inserted into the tree. This class would create the tree data structure consisting of several nodes of the BSTNode type.
- GenericsKbBSTApp – This class would be used to provide a menu to the user that they would make use of to load the knowledge base as well as add and search for data within the knowledge base. This class would create the binary search tree and load the given knowledge base to populate the binary search tree.
- GenericsKbArrayApp – This class contains all the code mentioned above but uses an array instead of a binary search tree. It has a separate node format within the class as it does not need to make use of the pointer system as the Binary search tree nodes require.

TESTING

Loading knowledge base:

```
Choose an action:
1. Load knowledge base from file
2. Add a new statement
3. Search by term
4. Search by term and sentence
5. Quit
Enter choice: 1
Enter file name: GenericsKB.txt
Knowledge base loaded successfully. 50000 entries processed.
```

Searching for existing data:

```
Choose an action:
1. Load knowledge base from file
2. Add a new statement
3. Search by term
4. Search by term and sentence
5. Quit
Enter choice: 3
Enter term to search: apple
Statement found: Apples are located in farmers markets. (Confidence: 1.0)
```

```
Choose an action:
1. Load knowledge base from file
2. Add a new statement
3. Search by term
4. Search by term and sentence
5. Quit
Enter choice: 3
Enter term to search: artificial intelligence
Statement found: Artificial intelligence is technology (Confidence: 1.0)
```

Searching for data that does not exist:

```
Choose an action:
1. Load knowledge base from file
2. Add a new statement
3. Search by term
4. Search by term and sentence
5. Quit
Enter choice: 3
Enter term to search: uct
Statement not found.
```

Search by term and sentence:

```
Choose an action:
1. Load knowledge base from file
2. Add a new statement
3. Search by term
4. Search by term and sentence
5. Quit
Enter choice: 4
Enter the term: criminologist
Enter the statement to search for: Criminologists are workers.
The statement was found and has a confidence score of 1.0.
```

Adding a term:

```
Choose an action:
1. Load knowledge base from file
2. Add a new statement
3. Search by term
4. Search by term and sentence
5. Quit
Enter choice: 2
Enter term: UCT
Enter sentence: UCT is a univeristy in Cape Town
Enter confidence score: 1.0
New statement added for term 'UCT'.
```

```
Choose an action:
1. Load knowledge base from file
2. Add a new statement
3. Search by term
4. Search by term and sentence
5. Quit
Enter choice: 3
Enter term to search: UCT
Statement found: UCT is a univeristy in Cape Town (Confidence: 1.0)
```

Additional coding outside basic requirements:

I had noticed that for my logic, the knowledge base would be discarded after each iteration which made me introduce a few checking procedures that would enable the user to know whether a knowledge base had been loaded or not before the any of the other functions could be carried out. In addition, the code is made in a way such that statements with a lower confidence level are not deleted from the knowledge base but are simply overlooked to save time, this enables only the information with the highest confidence levels to be presented to the user.

Git summary statistics:

```
abdul@AbdullahsPC:/mnt/c/Users/abdul/OneDrive - University of Cape Town/CSC2001F/Assignments/A
0; while read l; do echo $ln\: $l; ln=$((ln+1)); done) | (head -10; echo ...; tail -10)
0: commit 96d409139684d3229a238b2b8aab9846c4edaa77
1: Author: Abdullah <abdul@AbdullahsPC.>
2: Date: Sun Mar 16 13:36:53 2025 +0200
3:
4: Testing and Javadocs completed
5:
6: commit 147acc686d96f04a1e33ed3436d19cbc3db7dc48
7: Author: Abdullah <abdul@AbdullahsPC.>
8: Date: Sat Mar 15 17:32:59 2025 +0200
9:
...
25: Author: Abdullah <abdul@AbdullahsPC.>
26: Date: Sun Mar 9 21:50:00 2025 +0200
27:
28: Creating the node class and a bst class to insert nodes and/or search for nodes
29:
30: commit e48a23068aaa826f79f148d820eaebdde3a5b501
31: Author: Abdullah <abdul@AbdullahsPC.>
32: Date: Sun Mar 9 20:31:54 2025 +0200
33:
34: set up files for assignment 1
abdul@AbdullahsPC:/mnt/c/Users/abdul/OneDrive - University of Cape Town/CSC2001F/Assignments/A
```