Data Structures Project Diagram: Shehroz Khan (i19-1751), Ali Kamal (i19-1865), Razi Haider (i19-1762)

dataNode

- path: string
- line: int
- txtpath: string
- nextptr: dataNode*
- preptr: dataNode*

+ dataNode()
+ dataNode(int, string, string)
+ dataNode(const dataNode&)
+ gettxtpath(): string
+ ~dataNode()

template <class T> class file_reader

- CreateFolder(wchar_t* path): void
- readfile(int choice, DSDB<T>& node, int limit): void

my_data

- head: dataNode*
- tail: dataNode*
- count: int

+ my_data()
+ my_data(const my_data&)
+ get_head(): *dataNode
+ setnull(): void
+ operator = (const my_data obj): my_data
+ insert(int l, string p, string t): void
+ remove(int l, string p, string t): void
+ isEmpty(): bool
+ display(): string

<friend>

Composition

template class Node

- leaf: bool
- size: int
- data: my_data*
- key: T*
- child: Node<T>**
- next: Node<T>*
- next: Node<T>*

+ Node<T>()

<friend>

template <class T> class DSDB

- root: Node*

+ DSDB<T>()
+ check_for_same(T key, T* arr, int size): int
+ point_search(T x, int choice): void
+ range_search(T x, T x2, int choice): void
+ update(T key, int field, string old, string new_val): void
+ deletion(T x, int choice): void
+ insert(T x, int l, string p, string t): void
+ make_insertion(T x, my_data y, Node<T>* current, Node<T>* descendant): void
+ return_ancestor(Node<T>* current, Node<T>* findparentof): Node<T>*
+ getRoot(): Node<T>*
+ swap_dec(Node<T>** x, int j): inline void
+ swap_dec1d(T* x, int j): inline void
+ swap_dec1d_class(my_data* x, int j): inline void
+ send_same_idx(T* x, Node<T>* y, int i): inline void
+ send_same_idx_data(my_data* x, Node<T>* y, int i): inline void
+ send_same_idx_child(Node<T>** x, Node<T>* y, int i): inline void