Nolan Anderson

CPE 212-01 SP20

Josh Langford

02/07/2020

Project 03 Documentation

Classroom.cpp documentation

-Method name: Classroom()

-Returns: Function type constructor, does not have a return type.

-Pre condition: The class Classroom must be defined in the hpp file.

-Post condition: The Classroom class will have its objects initialized.

-Method name: AddStudent(const std::string &firstName, const std::string &lastName, unsigned int UID)

-Returns: boolean value

-Pre condition: There is a list defined and the student object must be defined.

-Post condition: The boolean truth value returned depends on whether or not a student has been added to the list.

-Method name: RemoveStudent(unsigned int UID)

-Returns: boolean value

-Pre condition: There is a list of students who have a unique ID.

Post condition: The boolean truth value returned depends on whether or not a student was removed from the list.

-Method name: AddProjects(const List<Assignment> &projects)

-Returns: Function does not return a type since it is void.

-Pre condition: There are defined project variables and a predefined list for each student object.

-Post condition: Each student's list will be updated with the projects that are called into the function.

-Method name: AddExams(const List<Assignment> &projects)

-Returns: Function does not return a type since it is void.

-Pre condition: There are defined exam values and there is a predefined list for each student object.

-Post condition: Each student's list will be updated with the exams that are called into the function.

-Method name: SetFinalExams(const List<Assignment> &projects)

-Returns: Function does not return a type since it is void.

-Pre condition: There is an integer that represents the final exam score and there is a predefined list for each student.

-Post condition: Each student will have a final exam score updated in their list.

<u>list_impl.hppdocumentation</u>

-Method name: List()

-Returns: Function type constructor, does not return a type.

-Pre condition: The class List is defined elsewhere.

-Post condition: The class List will have its objects initialized.

-Method name: ~List()

-Returns: Function type Deconstructor, does not return a type.

-Pre condition: The list values have either been initialized or declared.

-Post condition: All data within the list will be removed.

-Method name: AppendItem(const Type &data)

-Returns: Function type void, does not return a type.

-Pre condition: There is a predefined list and data.

-Post condition: The reference to the data called in will be appended to the end of the list.

-Method name: DeleteItem(const Type &data)

-Returns: boolean value

-Pre condition: There is a predefined list of items.

-Post condition: The boolean value returns true if the item was removed and false if it was not removed. _count is also updated.

-Method name: Count() const

-Returns: an unsigned int

-Pre condition: There is a predefined list.

-Post condition: The unsigned int returned is the size of the called in list.

-Method name: Front()

-Returns: a reference to a potentially varying data type.

-Pre condition: There is a list with an object defined at the front.

-Post condition: The reference data type returned depends on the data that is at the front of the list.

-Method name: Front() const

-Returns: a constant value of potentially varying data type.

-Pre condition: There is a list with an object defined at the front.

-Post condition: The const data type returned depends on the data that is at the front of the list.

-Method name: Back()

-Returns: a reference to a potentially varying data type.

-Pre condition: There is a list with an object defined at the end.

-Post condition: The reference data type returned depends on the data that is at the end of the list.

-Method name: Back() const

-Returns: a constant value of potentially varying data type.

-Pre condition: There is a list with an object defined at the end.

-Post condition: The const data type returned depends on the data that is at the end of the list.

-Method name: IterateItems() const

-Returns: a pointer to iterator type.

-Pre condition: There is a list with length > 1 with items that the iterator can point to.

-Post condition: The iterator will move to the next item in the list.

-Method name: AtEnd() const

-Returns: boolean value

-Pre condition: There is an iterator paired with a list.

-Post condition: The boolean value returned depends where the iterator is in the list.

-Method name: ResetIterator() const

-Returns: Function type void, does not return a type.

-Pre condition: There is an iterator value.

-Post condition: The iterator value will be returned to the beginning.