# CS 302 Project 1

Generated by Doxygen 1.8.6

Wed Jan 28 2015 22:08:13

# Contents

# Chapter 1

# Class Index

## 1.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

# Chapter 2

# File Index

## 2.1 File List

Here is a list of all files with brief descriptions:

# Chapter 3

# Class Documentation

## 3.1 Employee Class Reference

`#include <project1.h>`

### Public Member Functions

- void setName (string newName)
- void setEmployeeID (int newEmployeeID)
- void setAge (int newAge)
- void setSalary (double newSalary)
- string getName ()
- int getEmployeeID ()
- int getAge ()
- double getSalary ()

### 3.1.1 Member Function Documentation

#### 3.1.1.1 int Employee::getAge ( )

Retrieves the age data member from an employee object.

**Parameters**

| | |
|---|---|
| *None.* | |

**Precondition**

None, but caveat emptor - value could be temp or garbage if good data has not yet been entered.

**Postcondition**

The stored int in the age data member will be returned by the function.

**Returns**

Int stored in age data member.

#### 3.1.1.2 int Employee::getEmployeeID ( )

Retrieves the employeeID data member from an employee object.

**Parameters**

| | |
|---|---|
| *None.* | |

**Precondition**

None, but caveat emptor - value could be temp or garbage if good data has not yet been entered.

**Postcondition**

The stored int in the employeeID data member will be returned by the function.

**Returns**

Int stored in employeeID data member.

#### 3.1.1.3 string Employee::getName ( )

Retrieves the name data member from an employee object.

**Parameters**

| | |
|---|---|
| *None.* | |

**Precondition**

None, but caveat emptor - value could be temp or garbage if good data has not yet been entered.

**Postcondition**

The stored string in the name data member will be returned by the function.

**Returns**

String stored in name data member.

#### 3.1.1.4 double Employee::getSalary ( )

Retrieves the salary data member from an employee object.

**Parameters**

| | |
|---|---|
| *None.* | |

**Precondition**

None, but caveat emptor - value could be temp or garbage if good data has not yet been entered.

**Postcondition**

The stored double in the salary data member will be returned by the function.

**Returns**

Double stored in salary data member.

#### 3.1.1.5 void Employee::setAge ( int *newAge* )

Sets the age data member for an employee object. Use in conjunction with file read-in to load the employee data.

**Parameters**

| | |
|---|---|
| *newAge* | The age to be stored. |

**Precondition**

None.

**Postcondition**

The parameter int will be stored as the age data member of the object.

**Returns**

None.

#### 3.1.1.6 void Employee::setEmployeeID ( int *newEmployeeID* )

Sets the employeeID data member for an employee object. Use in conjunction with file read-in to load the employee data.

**Parameters**

| | |
|---|---|
| *newEmployeeID* | The employeeID to be stored. |

**Precondition**

None.

**Postcondition**

The parameter int will be stored as the employeeID data member of the object.

**Returns**

None.

#### 3.1.1.7 void Employee::setName ( string *newName* )

Sets the name data member for an employee object. Use in conjunction with file read-in to load the employee data.

**Parameters**

| | |
|---|---|
| *newName* | The name to be stored. |

**Precondition**

None.

**Postcondition**

The parameter string will be stored as the name data member of the object.

**Returns**

None.

#### 3.1.1.8 void Employee::setSalary ( double *newSalary* )

Sets the salary data member for an employee object. Use in conjunction with file read-in to load the employee data.

**Parameters**

| | |
|---|---|
| *newSalary* | The salary to be stored. |

**Precondition**

None.

**Postcondition**

The parameter double will be stored as the salary data member of the object.

**Returns**

None.

The documentation for this class was generated from the following files:

- project1.h
- project1.cpp

## 3.2 EmployeeContainer Class Reference

`#include <project1.h>`

### Public Member Functions

- int getNumEmployees ()
- double getAvgSalary ()
- int getAvgAge ()
- void sortByID ()
- bool addEmployee (Employee targetEmployee)
- bool exportEmployeeData ()

### 3.2.1 Member Function Documentation

#### 3.2.1.1 bool EmployeeContainer::addEmployee ( Employee *targetEmployee* )

Attempts to add a new employee object to the first empty element of the EmployeeContainer.

**Parameters**

| | |
|---|---|
| *targetEmployee* | The employee to be stored. |

**Precondition**

None.

**Postcondition**

If successful, the newEmployee object will be stored in the first empty element of the container and numEmployees will be incremented by 1. Otherwise, the container is unchanged and numEmployees remains the same.

**Returns**

True if the addition was successful, or false otherwise (eg if the container was full).

#### 3.2.1.2 bool EmployeeContainer::exportEmployeeData ( )

Prints stored employee information to console from the first stored element to the last. Note that if the data has been previously sorted, this will print employee's data to the console in order of ascending employeeID. If the data has not been sorted, employee data will be printed in the order in which it was entered into the container. In other words, in order to print sorted data, use the sortByID method before this one.

**Parameters**

| | |
|---|---|
| *None.* | |

**Precondition**

None.

**Postcondition**

If successful, the information will be printed to the console. Otherwise, nothing will be printed.

**Returns**

True if data was printed, or false otherwise (eg if the container was empty and there was nothing to print).

#### 3.2.1.3 int EmployeeContainer::getAvgAge ( )

Calculates the average age of the employee objects stored in the EmployeeContainer.

**Parameters**

| | |
|---|---|
| *None.* | |

**Precondition**

None.

**Postcondition**

The calculated age will be returned by the function. If the container is empty, the function will return -1.

**Returns**

The calculated average of the ages of the employees stored in the EmployeeContainer, or -1 if the container is empty.

#### 3.2.1.4 double EmployeeContainer::getAvgSalary ( )

Calculates the average of the salaries of the employee objects stored in the EmployeeContainer.

**Parameters**

| | |
|---|---|
| *None.* | |

**Precondition**

None.

**Postcondition**

The calculated value will be returned by the function. If the container is empty, the function will return -1.

**Returns**

The calculated average of the employee salaries stored in the EmployeeContainer, or -1 if the container is empty.

#### 3.2.1.5 int EmployeeContainer::getNumEmployees ( )

Returns the number of employee objects stored in the EmployeeContainer.

**Parameters**

| | |
|---|---|
| *None.* | |

**Precondition**

None.

**Postcondition**

The stored int in the numEmployees data member will be returned by the function. If the container is empty, the function will return 0.

**Returns**

Int stored in numEmployees data member.

#### 3.2.1.6 void EmployeeContainer::sortByID ( )

Sorts the employees within the EmployeeContainer in ascending order of employeeID in subsequent elements of the container (ie employee with the lowest employeeID in the first element, employee with the highest employeeID in the final used element).

**Parameters**

| | |
|---|---|
| *None.* | |

**Precondition**

None.

**Postcondition**

The employees will be sorted within the EmployeeContainer. If the number of employees is less than or equal to 1, no sorting can be done and no action will be performed.

**Returns**

None.

The documentation for this class was generated from the following files:

- project1.h
- project1.cpp

# Chapter 4

# File Documentation

## 4.1 project1.cpp File Reference

```
#include <iostream>
#include <fstream>
#include <string>
#include "project1.h"
```

### 4.1.1 Detailed Description

CS 302 Project 1 - code shell for use in employee data sorting/analyzing program

**Author**

Patrick Austin

**Date**

1/28/2015

## 4.2 project1.h File Reference

```
#include <string>
```

### Classes

- class Employee
- class EmployeeContainer

### 4.2.1 Detailed Description

CS 302 Project 1 - code shell for use in employee data sorting/analyzing program

**Author**

Patrick Austin

**Date**

1/28/2015

# Index