Software Design Document
for
Employee Record Database

Programming Assignment 1
January 29, 2020

Prepared By
Paul Abers
pa0034@uah.edu

Prepared For
Mr. James Williamson
CS 221, Data Structures in C++
Computer Science Department
University of Alabama in Huntsville

Contents

1 System Overview

The purpose of this assignment is to provide a simple and easy way to access an employee record class. The class must store an employee ID, employee name, department and annual salary of each employee. The class will eventually be ported to an employee database.

2 Referenced Documents

Programming Assignment 1 Statement of Work.

3 Architectural Design

3.1 Concept of Execution

This program creates a class structure to store information for an individual employee. The class stores an employee's first and last name, a unique employee ID, the employee's department ID and the employee's salary.

A database manager will have access to public get and set methods of the class in order to set the various attributes for the employee as well as get them later. There is also a default constructor that initializes the class as well as a constructor that handles all inputs being included. A quick and easy print will also be provided for quickly displaying all attributes of the class.

3.2 Abstract Data Type

The employee record structure is implemented with a class structure separated in a cpp and header file.

3.3 Code Outline

This program will consist of the following fiels: EmployeeRecord.h and EmployeeRecord.cpp.

Private Attributes:

- m_iEmployeeID – int value for employee id
- m_sLastName – character array of length 32 for last name
- m_sFirstName – character array of length 32 for first name
- m_iDeptId – int for department id
- m_dSalary – double for employee's salary

Public Methods:

- EmployeeRecord() – default constructor
- EmplyoeeRecord(int ID, char *fName, char *lName, int dept, double sal) – initialization constructor
- getID() – return int value of employee id
- setID(int ID) – set employee id
- getName(char* fName, char *lName) – copy employee's first and last name into pointers passed
- setName(char* fName, char *lName) – set employee's first and last name to pointers passed
- getDept(int& d) – get value of employee's department
- setDept(int d) – set value of employee's department

- getSalary(double *sal) – pointer function to get employee's salary

- setSalary(double sal) – set employee's salary

- printRecord() – prints to screen all data for employee's record

4 Detailed Design

4.1 Source File: EmployeeRecord.h and EmployeeRecord.cpp

4.1.1 Function: EmployeeRecord()

4.1.1.1 Purpose

This is the default constructor for the EmployeeRecord class.

4.1.1.2 Arguments

This default constructor takes no arguments.

4.1.1.3 Return Value

A constructor, therefore no value is returned.

4.1.1.4 Function Outline in Pseudocode

Set employee id to 0, last name to "", first name to "", department id to 0 and salary to 0.0.

4.1.1.5 Tracability

This function will fulfil requirement 2.2.2.1 of SOW by providing a default constructor for the employee record class.

4.1.2 Function: EmplyoeeRecord()

4.1.2.1 Purpose

This is the optional constructor to set all values passed into function.

4.1.2.2 Arguments

int employee id, character array pointer first name, character array pointer last name, int for department id, double for salary.

4.1.2.3 Return Value

None

4.1.2.4 Function Outline in Pseudocode

Set employee id to ID, copy passed character array for lName into m$_sLastName, copypassedcharacte$

4.1.2.5 Tracability

This function will fulfil requirement 2.2.2.2 of SOW by providing a default constructor for the employee record class.

4.1.3 Function: EmplyoeeRecord()

4.1.3.1 Purpose

This is the destructor for the employee record.

4.1.3.2 Arguments

None

4.1.3.3 Return Value

None

4.1.3.4 Function Outline in Pseudocode

Properly destruct the class. Clean up and deallocate memory initialized for pointers for the first and last name character arrays.

4.1.3.5 Tracability

This function will fulfil requirement 2.2.2.3 of SOW by providing a default constructor for the employee record class.

4.1.4 Function: getID()

4.1.4.1 Purpose

This function allows a user to get the private employee ID.

4.1.4.2 Arguments

None

4.1.4.3 Return Value

Int value stored for employee id.

4.1.4.4 Function Outline in Pseudocode

Return value of member stored employee ID.

4.1.4.5 Tracability

This function will partially fulfil requirement 2.2.2.4 of SOW by providing a default constructor for the employee record class.

4.1.5 Function: setID()

4.1.5.1 Purpose

This function allows a user to set the private employee ID.

4.1.5.2 Arguments

Int value to set the member stored employee id to.

4.1.5.3 Return Value

Void

4.1.5.4 Function Outline in Pseudocode

Set internal member variable for employee id to passed integer value.

4.1.5.5 Tracability

This function will partially fulfil requirement 2.2.2.4 of SOW by providing a default constructor for the employee record class.

4.1.6 Function: getName()

4.1.6.1 Purpose

This function allows a user to get the private employee first and last names.

4.1.6.2 Arguments

Pointer to character array first name, pointer to character array last name

4.1.6.3 Return Value

void

4.1.6.4 Function Outline in Pseudocode

Copy contents of internal member variable character arrays for first and last name into the character arrays passed into the function.

4.1.6.5 Tracability

This function will partially fulfil requirement 2.2.2.5 of SOW by providing a default constructor for the employee record class.

4.1.7 Function: setName()

4.1.7.1 Purpose

This function allows a user to set the private employee first name and last name character arrays.

4.1.7.2 Arguments

Pointer to character array first name, pointer to character array last name

4.1.7.3 Return Value

void

4.1.7.4 Function Outline in Pseudocode

Copy contents of passed pointer to character arrays of first name and last name into internal member variable character arrays for first and last name.

4.1.7.5 Tracability

This function will partially fulfil requirement 2.2.2.5 of SOW by providing a default constructor for the employee record class.

4.1.8 Function: getDept()

4.1.8.1 Purpose

This function allows a user to get the internal member value for department id.

4.1.8.2 Arguments

int reference variable

4.1.8.3 Return Value

void

4.1.8.4 Function Outline in Pseudocode

Copy contents of internal member value department id into int variable referenced by the function argument.

4.1.8.5 Tracability

This function will partially fulfil requirement 2.2.2.6 of SOW by providing a default constructor for the employee record class.

4.1.9 Function: setDept()

4.1.9.1 Purpose

This function allows a user to set the employee department id.

4.1.9.2 Arguments

Int for department id

4.1.9.3 Return Value

void

4.1.9.4 Function Outline in Pseudocode

Set internal value for department id equal to the passed int value.

4.1.9.5 Tracability

This function will partially fulfil requirement 2.2.2.6 of SOW by providing a default constructor for the employee record class.

4.1.10 Function: getSalary()

4.1.10.1 Purpose

This pointer function allows a user to get the employee's salary.

4.1.10.2 Arguments

Pointer double

4.1.10.3 Return Value

void

4.1.10.4 Function Outline in Pseudocode

Copy contents of member variable for employee salary to the pointer variable passed as a function argument.

4.1.10.5 Tracability

This function will partially fulfil requirement 2.2.2.7 of SOW by providing a default constructor for the employee record class.

4.1.11 Function: setSalary()

4.1.11.1 Purpose

This function allows a user to set the employee's salary.

4.1.11.2 Arguments

double for salary

4.1.11.3 Return Value

void

4.1.11.4 Function Outline in Pseudocode

Set member variable for salary equal to the passed double variable.

4.1.11.5 Tracability

This function will partially fulfil requirement 2.2.2.7 of SOW by providing a default constructor for the employee record class.

4.1.12 Function: printRecord()

4.1.12.1 Purpose

This function prints all info for employee record to the screen.

4.1.12.2 Arguments

None

4.1.12.3 Return Value

void

4.1.12.4 Function Outline in Pseudocode

Print all internal variable values to screen.

4.1.12.5 Tracability

This function will fulfil requirement 2.2.2.8 of SOW by providing a default constructor for the employee record class.