|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Programming Assignment 1**  **Posted: Thursday, 21Jan2020**  **Documents Due Date: Wed 29Jan2020**  **Source Code Due Date: Mon 10Feb2020**  **Statement of Work Programming Assignment 1** | | | | |
| **1.0 Overview** | | | | |
|  | Smith, Smithe, Smyth, Smythe, and Jones is a wholesale computer equipment supplier which sells computers, computer parts, and computer perpherials to a large number of retail stores throughout the United States. The company would like to create a database of all its traveling sales representatives. This database must be capable of maintaining the employee ID, employee name, department, and annual salary of each sales rep. The first phase of development for this database will be to create the EmployeeRecord class. | | | |
| **2.0 Requirements** | | | | |
|  | The student shall define, develop, document, prototype, test, and modify as required the software system. | | | |
|  | **2.1** This software system shall consist of one source file (.cpp) and one header file (.h) defining a C++ class which can be used to store information on one employee. | | | |
|  |  | **2.2** The header file, which shall be named **EmployeeRecord.h**, shall define a C++ class. The source file, which shall be named **EmployeeRecord.cpp**, shall implement all functions of the class. | | |
|  |  |  | **2.2.1** The class shall contain the following **private** variables: (1) an int called **m\_iEmployeeID**, (2) a character array of length 32 called **m\_sLastName**, (3) a character array of length 32 called **m\_sFirstName** (4) an int called **m\_iDeptID**, and (5) a double called **m\_dSalary**. Details of these variables are defined below. | |
|  |  |  |  | **2.2.1.1 m\_iEmployeeID**--This integer uniquely identifies this employee. This field will be used as the key when searching for an employee in later programming assignments. |
|  |  |  |  |  |
|  |  |  |  | **2.2.1.2 m\_sLastName/m\_sFirstName**--These two character arrays hold the name of the employee. First and last names may be up to 31 characters in length. These variables may not be implemented as *string* objects. |
|  |  |  |  |  |
|  |  |  |  | **2.2.1.3 m\_iDeptID**--This integer identifies the employee's department within the company. |
|  |  |  |  |  |
|  |  |  |  | **2.2.1.4 m\_dSalary**--This double holds the employee's annual salary. |
|  |  |  | **2.2.2** The class shall contain the following **public** functions: a constructor and destructor and get and set functions for each of the variables. These functions shall work as described in the following paragraphs. | |
|  |  |  |  | **2.2.2.1 EmployeeRecord()**--The default constructor shall set the member variables to the following initial values: m\_iEmployeeID = 0, m\_sLastName = "", m\_sFirstName = "", m\_iDeptID = 0, and m\_dSalary = 0.0. **Pay careful attention to the warning you will receive in class about how to set the values of strings implemented as character arrays. You may NOT substitute the *string* template for character arrays in this assignment.** |
|  |  |  |  | **2.2.2.2 EmployeeRecord(int ID, char \*fName, char \*lName, int dept, double sal)** --This constructor shall set the member variables to the values passed into the function. The arguments **fname** and **lName** are pointers to character arrays. |
|  |  |  |  | **2.2.2.3 ~EmployeeRecord()**--The destructor shall take care of cleaning up and deallocating any memory that pointers within this class may have reference to. |
|  |  |  |  | **2.2.2.4 int getID()/void setID(int ID)**--The function getID() shall return the int value stored in the member variable **m\_iEmployeeID**. The function setID() will set the member variable **m\_iEmployeeID** to the value of its' argument. |
|  |  |  |  | **2.2.2.5 void getName(char \*fName, char \*lName), void setName(char \*fName, char \*lName)** --The getName() function shall copy the member variables **m\_sFirstName** and **m\_sLastName** into the character arrays pointed to by the function arguments. The setName() function will copy the function arguments into the member variables **m\_sFirstName** and **m\_sLastName**. |
|  |  |  |  | **2.2.2.6 void getDept(int& d), void setDept(int d)**--The getDept() function shall be defined as a reference function. That is, a call to this function will copy the member variable **m\_iDeptID** into the int variable referenced by the function argument. The setDept() function will copy the function argument into the member variable **m\_iDeptID**. |
|  |  |  |  | **2.2.2.7 void getSalary(double \*sal), void setSalary(double sal)**-- The getSalary() function shall be defined as a pointer function. That is, a call to this function will copy the member variable **m\_dSalary** into the int variable pointed to by the function argument. The function setSalary() shall copy the function argument into the member variable **m\_dSalary**. |
|  |  |  |  | **2.2.2.8 void printRecord()**--This function shall print to the screen all data found in the employee's record. |
| **3.0 Deliverables** | | | | |
|  | These products shall be delivered to the instructor electronically via e-mail as specified below. **3.1 Design Document and Test Plan --** The student shall provide a filled out Sprint Report form for instructor approval NLT (Not Later Than) Wednesday, January 29 @ 11:59pm. **3.2 Program source files --** The student shall provide fully tested electronic copies of the .cpp and .h files. These files must be submitted to the instructor via e-mail. The files shall be delivered NLT Monday, February 10 @ 11:59pm. | | | |
| **4.0 Period of Performance** | | | | |
|  | The period of performance of this assignment is approximately 19 days from the date of assignment. | | | |