```
 1 ********************************************
 2 * PROGRAMMED BY : Saul Moreno
 3 * STUDENT ID    : 269491
 4 * CLASS         : CS 1B MW-7:30pm
 5 * ASSIGNMENT #2 : Employee-inheritance
 6 ********************************************
 7
 8 This Program will display the Data for the C1SCE Employee, The Programmers
 9 and Software Architects. It will display all other pertinent data
10
11 Data:
12 C1SCEmployees
13
14 These two examples are me just setting the employee's information automatically
15 Name           Employee's Id Phone         Age  Gender Job Title       Salary    Date
16 Tom Brady      12345         949-555-1234  42   M      Quarterback     $8000000   8/31/2018
17 Aaron Rogers   12346         310-555-5555  36   M      Quarterback     $777123    05/08/2019
18
19 This one is me manually changing the employee's information
20 Tom Rogers     354534        959-432-4325  69   F      Person          $0         12-12-12
21
22 Back to the information being added automatically
23
24 Oprah Winfrey  98765         730-703-1234  64   F      Talk Show Host  $9900000   12/25/2017
25 Sally Designer 77777         203-555-6789  36   M      Comedian        $500500    03/01/2012
26
27 Programmers:
28
29 Name           Employee's Id Phone         Age  Gender Job Title       Salary    Date
30 Sam Software   54321         819-123-4567  21   M      Programmer      $223000    12/24/2017
31 Mary Coder     65432         310-555-5555  28   F      Programmer      $770123    02/08/2019
32
33 Name           Department    Supervisor Name   Raise % C++ Knowledge   Java Knowledge
34 Sam Software   5432122       Joe Boss          4       Yes             No
35 Mary Coder     6543222       Mary Leader       7       Yes             Yes
36
37 SoftWare Architects
38
39 Name           Employee's Id Phone         Age  Gender Job Title       Salary    Date
40 Alex Arch      88888         959-353-3243  21   M      Architect       $231243    12/12/12
41 Sally Dsigner  87878         310-555-8888  38   F      Architect       $870123    12/12/13
42
43 This one is me manually changing the employee's information
44 Name           Department    Supervisor Name  Raise Years of Experience
45 Alex Arch      4545          Nobody Cares     -45   3
```

```cpp
/*****************************************************************************
 * AUTHOR      : Saul Moreno
 * STUDENT ID  : 269491
 * ASSIGNMENT# : 2
 * CLASS       : CS1C
 * SECTION     : MW 5:00pm
 * DUE DATE    : 1/29/2020
 *****************************************************************************/

#include "header.h"
#include "date.h"
#include "employee.h"
#include "programmer.h"
#include "softwareArchitect.h"
/*****************************************************************************
 * Employee-inheritance
 * ---------------------------------------------------------------------------
 * This program will output the class heading
 * ---------------------------------------------------------------------------
 * INPUT:
 *     <There is no input for this program - output data is obtained through
 *     the constant.>
 *
 * OUTPUT:
 *  <This program will output a class heading
 *****************************************************************************/
int main()
{
    Employee firstEmployee;    // variable that has access to the class
    Programmer programOne;     // variable that has access to the class
    SoftwareArchitect softOne; // variable that has access to the class

    //Calls the functions to print out class header
    PrintHeader("Employee-inheritance", 2, 'A');

    cout << "This Program will display the Data for the C1SCE Employee, The"
         << " Programmers" << endl
         << "and Software Architects. It will display all other pertinent data";
    cout << endl << endl;

    cout << "Data:" << endl << "C1SCEmployees\n\n";

    cout << "These two examples are me just setting the employee's information"
         << " automatically\n";
    //Calls function to display the table
    firstEmployee.DisplayTable();
    //Calls the function to set the initial values
    firstEmployee.SetInitial("Tom Brady",12345,"949-555-1234",42,'M',
                             "Quarterback",8000000,"8/31/2018");
    //Calls the functions to display the employee's information
    firstEmployee.Display();
    cout << endl;
    //Calls the function to set the initial values
    firstEmployee.SetInitial("Aaron Rogers", 12346, "310-555-5555",36,'M',
                             "Quarterback", 777123, "05/08/2019");
    //Calls the functions to display the employee's information
    firstEmployee.Display();
```

```
58
59      /*************************************************************************
60       * These functions call will change the values that the variables were
61       * set to
62       *************************************************************************/
63      cout << endl << endl;
64      cout << "This one is me manually changing the employee's information\n";
65      firstEmployee.ChangeEmployeesName("Tom Rogers");
66      firstEmployee.ChangeEmployeesId(354534);
67      firstEmployee.ChangeEmployeesPhone("959-432-4325");
68      firstEmployee.ChangeEmployeesAge(69);
69      firstEmployee.ChangeEmployeesGender('F');
70      firstEmployee.ChangeEmployeesJobTitle("Person");
71      firstEmployee.ChangeEmployeesSalary(0);
72      firstEmployee.ChangeEmployeesHireDate("12-12-12");
73
74      //Calls the functions to display the employee's information
75      firstEmployee.Display();
76      cout << endl;
77
78      cout << "\nBack to the information being added automatically\n\n";
79      firstEmployee.SetInitial("Oprah Winfrey", 98765, "730-703-1234",64,'F',
80                               "Talk Show Host", 9900000, "12/25/2017");
81      //Calls the functions to display the employee's information
82      firstEmployee.Display();
83      cout << endl;
84
85      //Calls the function to set the initial values
86      firstEmployee.SetInitial("Sally Designer", 77777, "203-555-6789",36,'M',
87                                  "Comedian", 500500, "03/01/2012");
88      //Calls the functions to display the employee's information
89      firstEmployee.Display();
90      cout << endl << endl;
91
92      cout << "Programmers:\n\n";
93
94      //Calls function to display the table
95      firstEmployee.DisplayTable();
96      //Calls the function to set the initial values
97      firstEmployee.SetInitial("Sam Software",54321,"819-123-4567",21,'M',
98                               "Programmer",223000,"12/24/2017");
99      //Calls the functions to display the employee's information
100     firstEmployee.Display();
101     cout << endl;
102     //Calls the function to set the initial values
103     firstEmployee.SetInitial("Mary Coder",65432,"310-555-5555",28,'F',
104                              "Programmer",770123,"02/08/2019");
105     //Calls the functions to display the employee's information
106     firstEmployee.Display();
107     cout << endl << endl;
108
109     //Calls function to display the table
110     programOne.DisplayTable();
111     //Calls the function to set the initial values
112     programOne.SetInitial("Sam Software", 5432122, "Joe Boss", 4, "Yes", "No");
113     //Calls the functions to display the employee's information
114     programOne.Display();
```

```
115     //Calls the function to set the initial values
116     programOne.SetInitial("Mary Coder", 6543222,"Mary Leader", 7, "Yes", "Yes");
117     //Calls the functions to display the employee's information
118     programOne.Display();
119     cout << endl;
120
121     cout << "SoftWare Architects\n\n";
122
123     //Calls function to display the table
124     firstEmployee.DisplayTable();
125     //Calls the function to set the initial values
126     firstEmployee.SetInitial("Alex Arch",88888,"959-353-3243",21,'M',"Architect",
127                              231243,"12/12/12");
128     //Calls the functions to display the employee's information
129     firstEmployee.Display();
130     cout << endl;
131     //Calls the function to set the initial values
132     firstEmployee.SetInitial("Sally Dsigner",87878,"310-555-8888",38,'F',
133                                "Architect", 870123,"12/12/13");
134     //Calls the functions to display the employee's information
135     firstEmployee.Display();
136
137     cout << endl << endl;
138
139     cout << "This one is me manually changing the employee's information\n";
140     //Calls function to display the table
141     softOne.DisplayTable();
142
143     /*************************************************************************
144     * These functions call will change the values that the variables were
145     * set to
146     *************************************************************************/
147     softOne.SetInitial("Alex Arch",543422,"Big Boss",6,4);
148     softOne.ChangeDepartNum(4545);
149     softOne.ChangeSuperName("Nobody Cares");
150     softOne.ChangeSalPer(-45);
151     softOne.ChangeNumExp(3);
152     //Calls the functions to display the employee's information
153     softOne.Display();
154
155     cout << endl << endl;
156
157     return 0;
158 }
159
160
161
```

```
1 /*****************************************************************************
2  * AUTHOR      : Saul Moreno
3  * STUDENT ID  : 269491
4  * ASSIGNMENT# : 2
5  * CLASS       : CS1C
6  * SECTION     : MW 5:00pm
7  * DUE DATE    : 1/29/2020
8  *****************************************************************************/
9
10 #ifndef HEADER_H_
11 #define HEADER_H_
12
13 #include <iostream>  // input and output
14 #include <iomanip>   // setprecision and setw
15 #include <string>    // allows to use strings
16 #include <limits>    //
17 #include <ios>       //
18 #include <fstream>   // file in & out
19 #include <time.h>    // system time
20 #include <stdlib.h>  // srand and rand
21 using namespace std;
22
23 void PrintHeader(string asName, int asNum, char asType);
24
25 #endif /* HEADER_H_ */
26
```

```cpp
/*****************************************************************************
 * AUTHOR       : Saul Moreno
 * STUDENT ID   : 269491
 * ASSIGNMENT# : 2
 * CLASS        : CS1C
 * SECTION      : MW 5:00pm
 * DUE DATE     : 1/29/2020
 *****************************************************************************/

#include"header.h"

/*****************************************************************************
*FUNCTION - PrintHeader
*_____________________________________________________________________________
*This function receives an assignment name, type and number then outputs the
* appropriate header - returns nothing.
*_____________________________________________________________________________
*PRE-CONDITIONS
*     asName:  Has to be previously defined
*     asType:    Has to be previously defined
*     asNum: Has to be previously defined
*
*POST-CONDITIONS
*      This function will output class heading.
*      <Post-conditions are the changed outputs either passed by value or
*      by reference OR anything affected by the function.
*
*****************************************************************************/

void PrintHeader(string asName,  // IN - Assignment Name
                 int asNum,  // IN - assignment type
                             // (LAB or ASSIGNMENT)
                 char asType)// IN - assignment number
{
 cout << left;
 cout << "*********************************************\n";
 cout << "* PROGRAMMED BY : Saul Moreno\n";
 cout << "* "<< setw (14) << "STUDENT ID" << ": 269491\n";
 cout << "* "<< setw (14) << "CLASS" << ": CS 1B MW-7:30pm\n";
 cout << "* ";
 if (toupper (asType) == 'L')
 {
  cout << "LAB #"<< setw(9);
 }
 else
 {
  cout <<"ASSIGNMENT #" << setw(2);
 }
 cout << asNum << ": " << asName << endl;
 cout << "*********************************************\n\n";
 cout <<right;
}

```

```
1 /*****************************************************************************
2  * AUTHOR      : Saul Moreno
3  * STUDENT ID  : 269491
4  * ASSIGNMENT# : 2
5  * CLASS       : CS1C
6  * SECTION     : MW 5:00pm
7  * DUE DATE    : 1/29/2020
8  *****************************************************************************/
9
10 #ifndef DATE_H_
11 #define DATE_H_
12
13 #include "header.h"
14
15 class Date
16 {
17     public:
18     Date();
19     ~Date();
20
21
22     void ChangeDate(int nMonth, int nDay, int nYear);//Changes the month, day
23                                                      // and year
24
25     private:
26     int month; // holds the value for the month
27     int day;   // holds the value for the day
28     int year;  // holds the value for the year
29 };
30
31 #endif /* DATE_H_ */
32
```

```cpp
/*****************************************************************************
 * AUTHOR       : Saul Moreno
 * STUDENT ID   : 269491
 * ASSIGNMENT# : 2
 * CLASS        : CS1C
 * SECTION      : MW 5:00pm
 * DUE DATE     : 1/29/2020
 *****************************************************************************/

#include "date.h"

Date::Date()
{
    month = 0;
    day   = 0;
    year  = 0;
}

Date::~Date()
{

}

/*****************************************************************************
*FUNCTION - ChangeDate
*_____________________________________________________________________________
*This function will change the employee's hire date.
*_____________________________________________________________________________
*PRE-CONDITIONS
* nMonth: has to be previously defined.
* nDay  : has to be previously defined.
* nYear : has to be previously defined.
*
*POST-CONDITIONS
* nMonth - sets the new value for employee's month.
* nDay   - sets the new value for employee's day.
* nYear  - sets the new value for employee's year.
*
*****************************************************************************/
void Date::ChangeDate(int nMonth, int nDay, int nYear)
{
    month = nMonth;
    day   = nDay;
    year  = nYear;

}

```

```
/*****************************************************************************
 * AUTHOR      : Saul Moreno
 * STUDENT ID  : 269491
 * ASSIGNMENT# : 2
 * CLASS       : CS1C
 * SECTION     : MW 5:00pm
 * DUE DATE    : 1/29/2020
 *****************************************************************************/

#ifndef EMPLOYEE_H_
#define EMPLOYEE_H_

#include "header.h"

class Employee
{
    public:
    /**************************
    ****CONSTRUCTOR/DESTRUCTOR**
    **************************/
    Employee();  //DEFAULT CONSTRUCTOR
    ~Employee(); //DEFAULT DESTRUCTOR

    /******************
    **** MUTATORS *****
    ******************/
    void SetInitial(string eName, int eId, string ePhone, int eAge,
                    char eGender, string eTitle, double eSalary, string eHire);
    void ChangeEmployeesName(string eName); //changes the employee's name
    void ChangeEmployeesId(int eId); //changes the employee's id
    void ChangeEmployeesPhone(string ePhone); //changes the employee's phone #
    void ChangeEmployeesAge(int eAge); //changes the employee's age
    void ChangeEmployeesGender(char eGender); //changes the gender
    void ChangeEmployeesJobTitle(string eTitle); //changes the employee's job
                                                 // title
    void ChangeEmployeesSalary(double eSalary); //changes the employee's salary
    void ChangeEmployeesHireDate(string eHire); //changes the employee's hire
                                                 // date

    /******************
    **** ACCESSORS ****
    ******************/
    void DisplayTable() const; // Will display table
    void Display() const; // Will display employee information

    protected:
    const int SET_NAME  = 15; // - column width for the name
    const int SET_ID    = 14; // - column width for the id
    const int SET_PHONE = 14; // - column width for the phone
    const int SET_AGE   = 5;  // - column width for the age
    const int SET_GEN   = 7;  // - column width for the gender
    const int SET_TITLE = 16; // - column width for the job title
    const int SET_SAL   = 10; // - column width for the salary
    const int SET_DATE  = 8;  // - column width for the date
    string employeeName;   // - holds employee's name

    private:
```

```
58      int    employeeId;     // - holds employee's id
59      string phoneNumber;    // - holds employee's phone number
60      int    employeeAge;    // - holds employee's age
61      char   employeeGender; // - holds employee's gender
62      string employeeTitle;  // - holds employee's title
63      double employeeSalary; // - holds employee's salary
64      string hireDate;       // - holds employee's hire date
65
66  };
67
68  #endif /* EMPLOYEE_H_ */
69
```

```cpp
/*****************************************************************************
 * AUTHOR       : Saul Moreno
 * STUDENT ID   : 269491
 * ASSIGNMENT# : 2
 * CLASS        : CS1C
 * SECTION      : MW 5:00pm
 * DUE DATE     : 1/29/2020
 *****************************************************************************/

#include "employee.h"

Employee::Employee()
{
    employeeName = "Saul Moreno";
    employeeId = 255;       // - sets employee's id
    phoneNumber = "523-545-5254";
    employeeAge = 0;      // - sets employee's age
    employeeGender = 'M'; // - sets employee's gender
    employeeTitle = " ";
    employeeSalary = 0;   // - holds employee's salary
    hireDate = "12/12/21";
}

Employee::~Employee()
{

}

/*****************************************************************************
*FUNCTION - SetInitial
*_____________________________________________________________________________
*This function set the initial values for the employee
*_____________________________________________________________________________
*PRE-CONDITIONS
* eName  : has to be previously defined.
* eId    : has to be previously defined.
* ePhone : has to be previously defined.
* eAge   : has to be previously defined.
* eGender: has to be previously defined.
* eTitle : has to be previously defined.
* eSalary: has to be previously defined.
* eHire  : has to be previously defined.
*POST-CONDITIONS
* employeeName  - sets the new value for employee's name.
* employeeId    - sets the new value for employee's id.
* phoneNumber   - sets the new value for employee's phone number.
* employeeAge   - sets the new value for employee's age.
* employeeGender - sets the new value for employee's gender.
* employeeTitle  - sets the new value for employee's title.
* employeeSalary - sets the new value for employee's salary
* hireDate       - sets the new value for employee's hire date.
*****************************************************************************/
void Employee::SetInitial(string eName, int eId, string ePhone, int eAge,
                    char eGender, string eTitle, double eSalary, string eHire)
{
    employeeName = eName;
    employeeId = eId;
```

```
58      phoneNumber = ePhone;
59      employeeAge = eAge;
60      employeeGender = eGender;
61      employeeTitle = eTitle;
62      employeeSalary = eSalary;
63      hireDate = eHire;
64  }
65
66  /*****************************************************************************
67  *FUNCTION - ChangeEmployeesName
68  *_____________________________________________________________________________
69  *This function will change the employee's name.
70  *_____________________________________________________________________________
71  *PRE-CONDITIONS
72  * eName: has to be previously defined.
73  *
74  *POST-CONDITIONS
75  * employeeName - sets the new value for employee's name.
76  *
77  *****************************************************************************/
78  void Employee::ChangeEmployeesName(string eName)
79  {
80      employeeName = eName;
81  }
82
83  /*****************************************************************************
84  *FUNCTION - ChangeEmployeesId
85  *_____________________________________________________________________________
86  *This function will change the employee's ID.
87  *_____________________________________________________________________________
88  *PRE-CONDITIONS
89  * eId: has to be previously defined.
90  *
91  *POST-CONDITIONS
92  * employeeId - sets the new value for the employee's ID.
93  *
94  *****************************************************************************/
95  void Employee::ChangeEmployeesId(int eId)
96  {
97      employeeId = eId;
98  }
99
100 /*****************************************************************************
101 *FUNCTION - ChangeEmployeesPhone
102 *_____________________________________________________________________________
103 *This function changes the employee's phone number.
104 *_____________________________________________________________________________
105 *PRE-CONDITIONS
106 * ePhone: has to be previously defined.
107 *
108 *POST-CONDITIONS
109 * phoneNumber - sets the new value for the employee's phone number.
110 *
111 *****************************************************************************/
112 void Employee::ChangeEmployeesPhone(string ePhone)
113 {
114     phoneNumber = ePhone;
```

```
}

/*****************************************************************************
*FUNCTION - ChangeEmployeesAge
*_____________________________________________________________________________
*This function changes the employee's age.
*_____________________________________________________________________________
*PRE-CONDITIONS
* eAge: has to be previously defined.
*
*POST-CONDITIONS
* employeeAge - sets the new value for the employee.
*
*****************************************************************************/
void Employee::ChangeEmployeesAge(int eAge)
{
    employeeAge = eAge;
}

/*****************************************************************************
*FUNCTION - ChangeEmployeesGender
*_____________________________________________________________________________
*This function will change the employee's gender.
*_____________________________________________________________________________
*PRE-CONDITIONS
* eGender: has to be previously defined.
*
*POST-CONDITIONS
* employeeGender - sets the new value for employee's gender.
*
*****************************************************************************/
void Employee::ChangeEmployeesGender(char eGender)
{
    employeeGender = eGender;
}

/*****************************************************************************
*FUNCTION - ChangeEmployeesJobTitle
*_____________________________________________________________________________
*This function will set employee's job title.
*_____________________________________________________________________________
*PRE-CONDITIONS
* eTitle: has to be previously defined.
*
*POST-CONDITIONS
* employeeTitle - stores the new value for employee's job title.
*
*****************************************************************************/
void Employee::ChangeEmployeesJobTitle(string eTitle)
{
    employeeTitle = eTitle;
}

/*****************************************************************************
*FUNCTION - ChangeEmployeesSalary
*_____________________________________________________________________________
*This function will change the employee's salary.
```

```
172 *_____________________________________________________________________________
173 *PRE-CONDITIONS
174 * eSalary: has to previously be defined.
175 *
176 *POST-CONDITIONS
177 * employeeSalary - stores the new value for employees salary.
178 *
179 ******************************************************************************/
180 void Employee::ChangeEmployeesSalary(double eSalary)
181 {
182     employeeSalary = eSalary;
183 }
184
185 /******************************************************************************
186 *FUNCTION - ChangeEmployeesHireDate
187 *_____________________________________________________________________________
188 *This function will change the employee's hire date.
189 *_____________________________________________________________________________
190 *PRE-CONDITIONS
191 * eHire: has to previously be defined.
192 *
193 *POST-CONDITIONS
194 * hireDate - stores the new value for the hire date.
195 *
196 ******************************************************************************/
197 void Employee::ChangeEmployeesHireDate(string eHire)
198 {
199     hireDate = eHire;
200 }
201
202 /******************************************************************************
203 *FUNCTION - DisplayTable
204 *_____________________________________________________________________________
205 *This function will print out the table heading.
206 *_____________________________________________________________________________
207 *PRE-CONDITIONS
208 * This is a void function
209 *
210 *POST-CONDITIONS
211 * This will not change anything member variables
212 *
213 ******************************************************************************/
214 void Employee::DisplayTable() const
215 {
216     #include <iomanip>
217     cout << left;
218
219     cout << setw(SET_NAME) << "Name" << setw(SET_ID) << "Employee's Id"
220          << setw(SET_PHONE) << "Phone" << setw(SET_AGE) << "Age"
221          << setw(SET_GEN) << "Gender" << setw(SET_TITLE) << "Job Title"
222          << setw(SET_SAL) << "Salary" << setw(SET_DATE) << "Date" << endl;
223     cout << right;
224 }
225
226 /******************************************************************************
227 *FUNCTION - Display
228 *_____________________________________________________________________________
```

```
229 *This function will print out all the information that was stored.
230 *_____________________________________________________________________
231 *PRE-CONDITIONS
232 * This is a void function
233 *
234 *POST-CONDITIONS
235 * This will not change anything member variables
236 *
237 ************************************************************************/
238 void Employee::Display() const
239 {
240     #include <iomanip>
241 
242     cout << left;
243     cout << setprecision(7);
244 
245     cout << setw(SET_NAME) << employeeName
246          << setw(SET_ID) << employeeId
247          << setw(SET_PHONE) << phoneNumber
248          << setw(SET_AGE) << employeeAge
249          << setw(SET_GEN) << employeeGender
250          << setw(SET_TITLE) << employeeTitle
251          << '$'<< setw(SET_SAL) << employeeSalary
252          << setw(SET_DATE) << hireDate;
253     cout << right;
254     cout << setprecision(6);
255 }
256 
```

```
1 /*****************************************************************************
2  * AUTHOR      : Saul Moreno
3  * STUDENT ID  : 269491
4  * ASSIGNMENT# : 2
5  * CLASS       : CS1C
6  * SECTION     : MW 5:00pm
7  * DUE DATE    : 1/29/2020
8  *****************************************************************************/
9
10 #ifndef PROGRAMMER_H_
11 #define PROGRAMMER_H_
12
13 #include "employee.h"
14
15 class Programmer: public Employee
16 {
17     public:
18     /**************************
19     ****CONSTRUCTOR/DESTRUCTOR**
20     **************************/
21     Programmer();  //DEFAULT CONSTRUCTOR
22     ~Programmer(); //DEFAULT DESTRUCTOR
23
24     /******************
25     **** MUTATORS *****
26     ******************/
27     void SetInitial(string eName, int dNumber, string sName, int pInc,
28                     string cIdn, string javId);
29     void ChangeDepartmentNum(int dNum); //changes the employee's department #
30     void ChangeSupervisorName(string sName); //changes the supervisors name
31     void ChangePercentInc(int pInc); //changes the employee's salary % inc
32     void ChangeCppIdentifier(string cIden); //status of C++ knowledge
33     void ChangeJavaIdentifier(string jIden); //status of Java knowledge
34
35     /******************
36     **** ACCESSORS ****
37     ******************/
38     void DisplayTable() const; // Will display table
39     void Display() const; // Will display employee information
40
41     private:
42     int departNumber;      // - sets employee's department name
43     string supervisorName; // - sets employee's supervisors name
44     int percentInc;        // - sets employee's percent increase
45     string cppIden;          // - sets employee's c++ knowledge
46     string javaIden;         // - sets employee's java knowledge
47
48 };
49
50 #endif /* PROGRAMMER_H_ */
51
```

```cpp
/*****************************************************************************
 * AUTHOR       : Saul Moreno
 * STUDENT ID   : 269491
 * ASSIGNMENT# : 2
 * CLASS        : CS1C
 * SECTION      : MW 5:00pm
 * DUE DATE     : 1/29/2020
 *****************************************************************************/

#include "programmer.h"

Programmer::Programmer()
{
    departNumber = 0;  // - sets employee's department name
    percentInc = 0;        // - sets employee's percent increase
    cppIden = 'Y';         // - sets employee's c++ knowledge
    javaIden = 'Y';
}

Programmer::~Programmer()
{

}

/*****************************************************************************
*FUNCTION - SetInitial
*_____________________________________________________________________________
*This function set the initial values for the employee
*_____________________________________________________________________________
*PRE-CONDITIONS
* dNumber: has to be previously defined.
* sName  : has to be previously defined.
* pInc   : has to be previously defined.
* cIdn   : has to be previously defined.
* javId  : has to be previously defined.
*POST-CONDITIONS
* departNumber   - sets the new value for employee's department number.
* supervisorName - sets the new value for employee's supervisors name.
* percentInc     - sets the new value for employee's salary % increase.
* cppIden        - sets the new value for employee's C++ knowledge status(Y/N).
* javaIden       - sets the new value for employee's Java knowledge status(Y/N).
*****************************************************************************/
void Programmer::SetInitial(string eName, int dNumber, string sName, int pInc,
                            string cIdn, string javId)
{
    Employee::ChangeEmployeesName(eName);
    departNumber = dNumber;
    supervisorName = sName;
    percentInc = pInc;
    cppIden = cIdn;
    javaIden = javId;
}

/*****************************************************************************
*FUNCTION - CdepartmentNum
*_____________________________________________________________________________
*This function changes the employee's department number.
```

```
58 *_____________________________________________________________________________
59 *PRE-CONDITIONS
60 * dNum: has to be previously defined.
61 *
62 *POST-CONDITIONS
63 * departmentNumber- sets the new value for the employee's department number.
64 *
65 *****************************************************************************/
66 void Programmer::ChangeDepartmentNum(int dNum)
67 {
68     departNumber = dNum;
69 }
70
71 /*****************************************************************************
72 *FUNCTION - CsupervisorName
73 *_____________________________________________________________________________
74 *This function changes the employee's supervisors name.
75 *_____________________________________________________________________________
76 *PRE-CONDITIONS
77 * sName: has to be previously defined.
78 *
79 *POST-CONDITIONS
80 * supervisorName - sets the new value for the employee's supervisor.
81 *
82 *****************************************************************************/
83 void Programmer::ChangeSupervisorName(string sName)
84 {
85     supervisorName = sName;
86 }
87
88 /*****************************************************************************
89 *FUNCTION - CpercentInc
90 *_____________________________________________________________________________
91 *This function changes the employee's salary percent increase.
92 *_____________________________________________________________________________
93 *PRE-CONDITIONS
94 * pInc: has to be previously defined.
95 *
96 *POST-CONDITIONS
97 * percentInc- sets the new value for the employee's percent increase.
98 *
99 *****************************************************************************/
100 void Programmer::ChangePercentInc(int pInc)
101 {
102     percentInc = pInc;
103 }
104
105 /*****************************************************************************
106 *FUNCTION - CcppIdentifier
107 *_____________________________________________________________________________
108 *This function changes the employee's c++ status to (Y/N).
109 *_____________________________________________________________________________
110 *PRE-CONDITIONS
111 * cIdent: has to be previously defined.
112 *
113 *POST-CONDITIONS
114 * cppIden- sets a (Y/N) value.
```

```
115 *
116 ******************************************************************************/
117 void Programmer::ChangeCppIdentifier(string cIden)
118 {
119     cppIden = cIden;
120 }
121
122 /******************************************************************************
123 *FUNCTION - ChangeEmployeesPhone
124 *_____________________________________________________________________________
125 *This function changes the employee's java status (Y/N).
126 *_____________________________________________________________________________
127 *PRE-CONDITIONS
128 * jIden: has to be previously defined.
129 *
130 *POST-CONDITIONS
131 * javaIden- sets a (Y/N) value.
132 *
133 ******************************************************************************/
134 void Programmer::ChangeJavaIdentifier(string jIden)
135 {
136     javaIden = jIden;
137 }
138
139 /******************************************************************************
140 *FUNCTION - DisplayTable
141 *_____________________________________________________________________________
142 *This function will print out the table heading.
143 *_____________________________________________________________________________
144 *PRE-CONDITIONS
145 * This is a void function
146 *
147 *POST-CONDITIONS
148 * This will not change anything member variables
149 *
150 ******************************************************************************/
151 void Programmer::DisplayTable() const
152 {
153     #include <iomanip>
154     cout << left;
155
156     cout << setw(SET_NAME) << "Name" << setw(SET_ID) << "Department"
157          << setw(18) << "Supervisor Name" << setw(SET_DATE) << "Raise %"
158          << setw(SET_TITLE) << "C++ Knowledge" << setw(SET_ID)
159          << "Java Knowledge" << endl;
160     cout << right;
161 }
162
163 /******************************************************************************
164 *FUNCTION - Display
165 *_____________________________________________________________________________
166 *This function will print out all the information that was stored.
167 *_____________________________________________________________________________
168 *PRE-CONDITIONS
169 * This is a void function
170 *
171 *POST-CONDITIONS
```

```
172 * This will not change anything member variables
173 *
174 ****************************************************************************/
175 void Programmer::Display() const
176 {
177     string eName;
178     //Employee::Display();
179     //Employee::ChangeEmployeesName(eName);
180 #include <iomanip>
181     cout << left;
182
183     cout << setw(SET_NAME) << employeeName << setw(SET_ID) << departNumber
184          << setw(18) << supervisorName << setw(SET_DATE) << percentInc
185          << setw(SET_TITLE) << cppIden << setw(SET_ID)
186          << javaIden << endl;
187     cout << right;
188
189 }
190
```

```
1 /*****************************************************************************
2  * AUTHOR      : Saul Moreno
3  * STUDENT ID  : 269491
4  * ASSIGNMENT# : 2
5  * CLASS       : CS1C
6  * SECTION     : MW 5:00pm
7  * DUE DATE    : 1/29/2020
8  *****************************************************************************/
9
10 #ifndef SOFTWAREARCHITECT_H_
11 #define SOFTWAREARCHITECT_H_
12
13 #include "employee.h"
14
15 class SoftwareArchitect: public Employee
16 {
17     public:
18     /**************************
19     ****CONSTRUCTOR/DESTRUCTOR**
20     **************************/
21     SoftwareArchitect();  //DEFAULT CONSTRUCTOR
22     ~SoftwareArchitect(); //DEFAULT DESTRUCTOR
23
24     /******************
25     **** MUTATORS *****
26     ******************/
27     void SetInitial(string eName, int dNum, string sName, int sPer,
28                     int nExp); //sets the employee's value
29     void ChangeDepartNum(int dNum); // changes department name
30     void ChangeSuperName(string sName); // changes supervisors name
31     void ChangeSalPer(int sPer); //changes salary percent raise
32     void ChangeNumExp(int nExp); //changes employees's years of experience
33     /******************
34     **** ACCESSORS ****
35     ******************/
36     void DisplayTable() const; // Will display table heading
37     void Display() const; // Will display employee information
38
39     private:
40     int departmentNumber;  // - sets employee's department name
41     string supervisorName; // - sets employee's supervisors name
42     int salaryPercentInc;  // - sets the salary % increase
43     int numExp;            // - sets the number of experience
44 };
45
46 #endif /* SOFTWAREARCHITECT_H_ */
47
```

```cpp
/*****************************************************************************
 * AUTHOR       : Saul Moreno
 * STUDENT ID   : 269491
 * ASSIGNMENT# : 2
 * CLASS        : CS1C
 * SECTION      : MW 5:00pm
 * DUE DATE     : 1/29/2020
 *****************************************************************************/

#include "softwareArchitect.h"

SoftwareArchitect::SoftwareArchitect()
{
    departmentNumber = 0;
    salaryPercentInc = 0;
    numExp           = 0;
}

SoftwareArchitect::~SoftwareArchitect()
{

}

/*****************************************************************************
*FUNCTION - SetInitial
*_____________________________________________________________________________
*This function set the initial values for the employee
*_____________________________________________________________________________
*PRE-CONDITIONS
* eName : has to be previously defined.
* dNum  : has to be previously defined.
* sName : has to be previously defined.
* sPer  : has to be previously defined.
* nExp  : has to be previously defined.
*POST-CONDITIONS
* employeeName     - sets the new value for employee's name.
* departNumber     - sets the new value for employee's department number.
* supervisorName   - sets the new value for employee's supervisors name.
* salaryPercentInc - sets the new value for employee's salary % increase.
* numExp           - sets the new value for employee's # of experience.
*****************************************************************************/
void SoftwareArchitect::SetInitial(string eName, int dNum, string sName,
                                   int sPer, int nExp)
{
    Employee::ChangeEmployeesName(eName);
    departmentNumber = dNum;
    supervisorName = sName;
    salaryPercentInc = sPer;
    numExp = nExp;
}

/*****************************************************************************
*FUNCTION - ChangeDepartNum
*_____________________________________________________________________________
*This function changes the employee's department number.
*_____________________________________________________________________________
*PRE-CONDITIONS
```

```
58 * dNum: has to be previously defined.
59 *
60 *POST-CONDITIONS
61 * departNumber- sets the new value for employee's department number.
62 *
63 ***************************************************************************/
64 void SoftwareArchitect::ChangeDepartNum(int dNum)
65 {
66     departmentNumber = dNum;
67 }
68
69 /***************************************************************************
70 *FUNCTION - ChangeSuperName
71 *_________________________________________________________________________
72 *This function change the employee's supervisors name.
73 *_________________________________________________________________________
74 *PRE-CONDITIONS
75 * sName: has to be previously defined.
76 *
77 *POST-CONDITIONS
78 * supervisorName- sets the new value for employee's supervisors name.
79 *
80 ***************************************************************************/
81 void SoftwareArchitect::ChangeSuperName(string sName)
82 {
83     supervisorName = sName;
84 }
85
86 /***************************************************************************
87 *FUNCTION - ChangeSalPer
88 *_________________________________________________________________________
89 *This function will change the employee's salary percent increase.
90 *_________________________________________________________________________
91 *PRE-CONDITIONS
92 *sName: has to be previously defined.
93 *
94 *POST-CONDITIONS
95 *supervisorName- sets the new value for employee's supervisors name.
96 *
97 ***************************************************************************/
98 void SoftwareArchitect::ChangeSalPer(int sPer)
99 {
100     salaryPercentInc = sPer;
101 }
102
103 /***************************************************************************
104 *FUNCTION - ChangeNumExp
105 *_________________________________________________________________________
106 *This function will change the number of years of experience that the
107 *employee has.
108 *_________________________________________________________________________
109 *PRE-CONDITIONS
110 * nExp: has to be previously defined.
111 *
112 *POST-CONDITIONS
113 * numExp - sets the new value for employee's # of experience.
114 *
```

```
115 ******************************************************************************/
116 void SoftwareArchitect::ChangeNumExp(int nExp)
117 {
118     numExp = nExp;
119 }
120
121 /******************************************************************************
122 *FUNCTION - DisplayTable
123 *_____________________________________________________________________________
124 *This function will print out the table heading.
125 *_____________________________________________________________________________
126 *PRE-CONDITIONS
127 * This is a void function
128 *
129 *POST-CONDITIONS
130 * This will not change anything member variables
131 *
132 ******************************************************************************/
133 void SoftwareArchitect::DisplayTable() const
134 {
135     #include <iomanip>
136     cout << left;
137
138     cout << setw(SET_NAME) << "Name" << setw(SET_ID) << "Department"
139          << setw(17) << "Supervisor Name" << setw(6) << "Raise"
140          << setw(SET_GEN) << "Years of Experience" << endl;
141     cout << right;
142 }
143
144 /******************************************************************************
145 *FUNCTION - Display
146 *_____________________________________________________________________________
147 *This function will print out all the information that was stored.
148 *_____________________________________________________________________________
149 *PRE-CONDITIONS
150 * This is a void function
151 *
152 *POST-CONDITIONS
153 * This will not change anything member variables
154 *
155 ******************************************************************************/
156 void SoftwareArchitect::Display() const
157 {
158     #include <iomanip>
159     cout << left;
160
161     cout << setw(SET_NAME) << employeeName << setw(SET_ID) << departmentNumber
162          << setw(17) << supervisorName << setw(6) << salaryPercentInc
163          << setw(SET_GEN) << numExp << endl;
164     cout << right;
165 }
166
```