```
 1
 2 #ifndef MYHEADER_H_
 3 #define MYHEADER_H_
 4
 5 //Preprocessor directives
 6
 7 #include<iostream> //for input and output
 8 #include<iomanip>  //for output style
 9 #include<string>   //for using string
10
11 //using the name space standard
12 using namespace std;
13
14
15 //class date: for defining the date
16 class date
17 {
18 //public parts containing the method functions of the class
19 public:
20
21     //default constructor
22     date();
23
24     //destructor
25     ~date();
26
27 //protected attributes of the class (accessible by derived classes)
28 protected:
29     int month;  //PROCESS - for storing month
30     int day;//PROCESS - for storing day
31     int year;   //PROCESS - for storing year
32 };
33
34
35 //Class employee(derived from the class date):
36 //for setting and changing the attributes about the employees
37 class employee: protected date
38 {
39 //public parts containing the method functions of the class
40 public:
41     //default constructor
42     employee();
43
44
45     //destructor
46     ~employee();
47
48     //method function for setting the name
49     void setName(string empName);
50
51     //method function for setting the Id
52     void setId(string empId);
53
54     //method function for setting the phone number
55     void setPhoneNumber(string Number);
```

```
56
57      //method function for setting the age
58      void setAge(int empAge);
59
60      //method function for setting the gender
61      void setGender(char sex);
62
63      //method function for setting the job title
64      void setJobTitle(string title);
65
66      //method function for setting the salary
67      void setSalary(double income);
68
69      //method function for setting the hire date
70      void setDate(int startDay, int startMonth, int startYear);
71
72      //method function for printing the attributes of the class
73      void print();
74
75  //protected attributes of the class (accessible by derived classes)
76  protected:
77      string  name;       //PROCESS - storing the name
78      string  id;         //PROCESS - storing the id number
79      string  phoneNum;   //PROCESS - storing the phone number
80      int     age;    //PROCESS - storing the age
81      chargender;     //PROCESS - storing the gender
82      string  jobTitle;   //PROCESS - storing the job title
83      double  salary;     //PROCESS - storing the salary
84      datestartDate;  //PROCESS - storing the hire date
85  };
86
87  //class programmer(derived from the class employee):
88  //for setting and changing the attributes about the programmer
89  class programmer: protected employee
90  {
91  //public parts containing the method functions of the class
92  public:
93      //default constructor
94      programmer();
95
96
97      //destructor
98      ~programmer();
99
100     //method for setting the department number
101     void setDepartmentNum(string depNum);
102
103     //method for setting the supervisor's name
104     void setSupName(string supervisor);
105
106     //method for setting the salary increase percentage
107     void setPercentage(int percent);
108
109     //method for setting if the person knows c++
110     void cppIdentifier(bool cppIdent);
```

```
111
112     //method for setting if the person knows java
113     void javaIdentifier(bool javaIdent);
114
115     //method for printing the attributes of the class
116     void printProgrammer();
117
118 //private part of the class; containing the attributes of the class
119 private:
120     string  depNumber;      //PROCESS - storing the department number
121     string  supName;    //PROCESS - storing the supervisor's name
122     float   percSalaryInc;  //PROCESS - storing the salary increase percentage
123     boolcpp;        //PROCESS - storing if the person knows c++
124     booljava;           //PROCESS - storing if the person knows java
125
126 };
127
128 //class softArch(derived from the class employee):
129 //for setting and changing the attributes about the software Architects
130 class softArch: protected employee
131 {
132 //public parts containing the method functions of the class
133 public:
134
135     //default constructor
136     softArch();
137
138     //destructor
139     ~softArch();
140
141     //method for setting the department number
142     void setDepartmentNum(string departmnet);
143
144     //method for setting the supervisors name
145     void setSupName(string supervisor);
146
147     //method for setting the salary increase percentage
148     void setPercentage(float percent);
149
150     //method for setting the experience
151     void setExperience(int empYears);
152
153     //method for printing the attributes of the class
154     void printSoftArch();
155
156 //private part of the class; containing the attributes of the class
157 private:
158     string  depNumber;      //PROCESS - storing the department number
159     string  supName;    //PROCESS - storing the supervisor's name
160     float   percSalaryInc;  //PROCESS - storing the salary increase percentage
161     int     experience;     //PROCESS - storing the experience
162 };
163
164
165
```

```
166 #endif /* MYHEADER_H_ */
167
168
169
```

```cpp
/*****************************************************************************
 * PROGRAMMER : Ali Eshghi
 * STUDENT ID : 1112261
 * CLASS      : CS1C
 * SECTION    : MW 5pm
 * Assign #1  : CS1C corporation
 * DUE DATE   : 29 January 2020
 *****************************************************************************/

#include "MyHeader.h"

/*************************************************************
 * CS1C Corporation
 * ___________________________________________________________
 * This program prints out the data from the list of the
 * employee of the corporation using the classes method and
 * Inheritance of the classes and passing the data through
 * the method functions of the class
 * ___________________________________________________________
 * INPUT: N/A
 *
 * OUTPUT: table of the employees with their information
 *         (Name, Id, Phone #, Age, Gender, Job title,
 *         Salary, Hire date) and then the programmers
 *         information, same as the employees (with the
 *         additional information of Department #,
 *         supervisor's name, Raise Increase %, C++
 *         knowledge, and Java knowledge) and the Software
 *         Architect with the same information but
 *
 *
 *************************************************************/

int main()
{
    //Variables

    date    date;       //PROCESS - date class type variable
    employee    employee;   //PROCESS - employee class type variable
    programmer  programmer; //PROCESS - programmer class type variable
    softArcharchitect;  //PROCESS - softArch class type variable



    cout << "Data:" << endl;
    cout << "CS1C Employees" << endl;

    cout << left;
    cout << setw(15)  << "Name"      << setw(9) << "ID"    << setw(15)
         << "Phone #" << setw(7)     << "Age"    << setw(9) << "Gender"
         << setw(15)  << "Job title" << setw(15) << "Salary"
         << setw(15)  << "Hire date" << endl;
    cout << "-----------------------------------------------";
    cout << "-----------------------------------------------";
    cout << endl;

```

```
57      //passing the employees information
58      //to the methods to set the data
59
60      employee.setName("Tom Brady");
61      employee.setId("12345");
62      employee.setPhoneNumber("949-555-1234");
63      employee.setAge(42);
64      employee.setGender('M');
65      employee.setJobTitle("Quarterback");
66      employee.setSalary(8000000);
67      employee.setDate(31,8,2018);
68
69      //printing the information
70      employee.print();
71
72      //passing the employees information
73      //to the methods to set the data
74
75      employee.setName("Aaron Rodgers");
76      employee.setId("12346");
77      employee.setPhoneNumber("310-555-5555");
78      employee.setAge(36);
79      employee.setGender('M');
80      employee.setJobTitle("Quarterback");
81      employee.setSalary(770123);
82      employee.setDate(8,5,2019);
83
84      //printing the information
85      employee.print();
86
87      //passing the employees information
88      //to the methods to set the data
89
90      employee.setName("Oprah Winfrey");
91      employee.setId("98765");
92      employee.setPhoneNumber("703-703-1234");
93      employee.setAge(64);
94      employee.setGender('F');
95      employee.setJobTitle("Talk show Host");
96      employee.setSalary(9900000);
97      employee.setDate(25,12,2017);
98
99      //printing the information
100     employee.print();
101
102     //passing the employees information
103     //to the methods to set the data
104
105     employee.setName("Jay Leno");
106     employee.setId("77777");
107     employee.setPhoneNumber("203-555-6789");
108     employee.setAge(69);
109     employee.setGender('M');
110     employee.setJobTitle("Comedian");
111     employee.setSalary(500500);
112     employee.setDate(1,3,2012);
```

```
113
114     //printing the information
115     employee.print();
116
117     cout << endl << endl;
118
119
120     cout << "Programmers:" << endl;
121
122     //passing the programmers information
123     //to the employees methods to set the data
124
125     employee.setName("Sam Software");
126     employee.setId("54321");
127     employee.setPhoneNumber("819-123-4567");
128     employee.setAge(21);
129     employee.setGender('M');
130     employee.setJobTitle("Programmer");
131     employee.setSalary(223000);
132     employee.setDate(24,12,2017);
133
134     //passing the programmers information
135     //to the methods to set the data
136
137     programmer.setDepartmentNum("5432122");
138     programmer.setSupName("Joe Boss");
139     programmer.setPercentage(4);
140     programmer.cppIdentifier(true);
141     programmer.javaIdentifier(false);
142
143     cout << left;
144     cout << setw(15)  << "Name"      << setw(9) << "ID"    << setw(15)
145          << "Phone #" << setw(7)     << "Age"    << setw(9) << "Gender"
146          << setw(15)  << "Job title" << setw(15) << "Salary"
147          << setw(15)  << "Hire date" << endl;
148     cout << "-------------------------------------------------";
149     cout << "-------------------------------------------------";
150     cout << endl;
151
152     //printing the employee informations of the programmer
153     employee.print();
154
155     //passing the programmers information
156     //to the employees methods to set the data
157
158     employee.setName("Mary Coder");
159     employee.setId("65432");
160     employee.setPhoneNumber("310-555-5555");
161     employee.setAge(28);
162     employee.setGender('F');
163     employee.setJobTitle("Programmer");
164     employee.setSalary(770123);
165     employee.setDate(8,2,2019);
166
167     //printing the employee informations of the programmer
168     employee.print();
```

```
169
170     cout << endl << endl;
171
172     cout << left;
173     cout << setw(14)  << "Name"      << setw(14) << "Department"    << setw(22)
174          << "Supervisor's Name" << setw(12)     << "Raise %"    << setw(19) << "C++
    Knowledge"
175          << "Java Knowledge" << endl;
176     cout << "-----------------------------------------------";
177     cout << "-----------------------------------------------";
178     cout << endl;
179
180     //printing programmers information
181     employee.setName("Sam Software");
182     programmer.printProgrammer();
183
184     //passing the programmers information
185     //to the methods to set the data
186
187     programmer.setDepartmentNum("6543222");
188     programmer.setSupName("Mary Leader");
189     programmer.setPercentage(7);
190     programmer.cppIdentifier(true);
191     programmer.javaIdentifier(true);
192
193
194     //printing programmers information
195     employee.setName("Mary Coder");
196     programmer.printProgrammer();
197
198     cout << endl << endl;
199
200
201     cout << "Software Architects:" << endl;
202
203     //passing the Software Architect information
204     //to the employees methods to set the data
205
206     employee.setName("Alex Arch");
207     employee.setId("88888");
208     employee.setPhoneNumber("819-123-4444");
209     employee.setAge(31);
210     employee.setGender('M');
211     employee.setJobTitle("Architect");
212     employee.setSalary(323000);
213     employee.setDate(24,12,2018);
214
215     //passing the Software Architect information
216     //to the methods to set the data
217
218     architect.setDepartmentNum("5434222");
219     architect.setSupName("Big Boss");
220     architect.setPercentage(5);
221     architect.setExperience(4);
222
223     cout << left;
```

```
224     cout << setw(15)  << "Name"      << setw(9) << "ID"    << setw(15)
225          << "Phone #" << setw(7)     << "Age"    << setw(9) << "Gender"
226          << setw(15)  << "Job title" << setw(15) << "Salary"
227          << setw(15)  << "Hire date" << endl;
228     cout << "-------------------------------------------------";
229     cout << "-------------------------------------------------";
230     cout << endl;
231
232     //printing the employee informations of the software Architect
233     employee.print();
234
235     //passing the Software Architect information
236     //to the employees methods to set the data
237
238     employee.setName("Sally Designer");
239     employee.setId("87878");
240     employee.setPhoneNumber("310-555-8888");
241     employee.setAge(38);
242     employee.setGender('F');
243     employee.setJobTitle("Architect");
244     employee.setSalary(870123);
245     employee.setDate(8,2,2013);
246
247     //printing the employee informations of the software Architect
248     employee.print();
249
250     cout << endl << endl;
251
252     cout << left;
253     cout << setw(13)  << "Name"   << setw(15)    << "Department" << setw(22)
254          << "Supervisor's Name"   << setw(12)     << "Raise %"
255          << "Years of experience" << endl;
256     cout << "-------------------------------------------------";
257     cout << "-------------------------------------------------";
258     cout << endl;
259
260     //printing programmers information
261     employee.setName("Alex Arch");
262     architect.printSoftArch();
263
264     //passing the Software Architect information
265     //to the methods to set the data
266
267     architect.setDepartmentNum("6543422");
268     architect.setSupName("Big Boss");
269     architect.setPercentage(8);
270     architect.setExperience(11);
271
272     //printing programmers information
273     employee.setName("Sally Designer");
274     architect.printSoftArch();
275
276
277
278
279     return 0;
```

```
}
```

```cpp
/*****************************************************************************
 * PROGRAMMER : Ali Eshghi
 * STUDENT ID : 1112261
 * CLASS      : CS1C
 * SECTION    : MW 5pm
 * Assign #1  : CS1C corporation
 * DUE DATE   : 29 January 2020
 *****************************************************************************/

#include "MyHeader.h"

/*****************************************************************************
 * Methods for class date
 *****************************************************************************/


//non-Default constructor
date::date()
{
    //INITIALIZATION
    day   = 0;
    month = 0;
    year  = 0;
}

//destruactor
date::~date() {}


/*****************************************************************************
 * Methods for class employee
 *****************************************************************************/

//default constructor
employee::employee()
{
    //INITIALIZATION
    name.clear();
    id.clear();
    phoneNum.clear();
    jobTitle.clear();

    age    = 0;
    salary = 0;

    gender = ' ';

}

//destructor
employee::~employee() {}

//method for getting the name from the client and store it in name attribute
void employee::setName(string empName)
{
```

```
 56     name = empName;
 57 }
 58
 59 //methpod for getting the id from the client and store it in id attribute
 60 void employee::setId(string empId)
 61 {
 62     id = empId;
 63 }
 64
 65 //method for getting the phone number from the client and store it in
 66 //the phoenNum attribute
 67 void employee::setPhoneNumber(string number)
 68 {
 69     phoneNum = number;
 70 }
 71
 72 //method for getting the age from the client and store it in
 73 //age attribute
 74 void employee::setAge(int empAge)
 75 {
 76     age = empAge;
 77 }
 78 //method for getting the gender from the client and store it
 79 //in gender attribute
 80 void employee::setGender(char sex)
 81 {
 82     gender = sex;
 83 }
 84
 85 //method for getting the job title from the client and store
 86 //it in the jobTile attribute
 87 void employee::setJobTitle(string title)
 88 {
 89     jobTitle = title;
 90 }
 91
 92 //method for getting the salary from the client and store it
 93 //in salary attribute
 94 void employee::setSalary(double income)
 95 {
 96     salary = income;
 97 }
 98
 99 //method for getting the hire date attributes and save the date
100 //into the attributes of day, month, and year
101 void employee::setDate(int startDay, int startMonth, int startYear)
102 {
103     day   = startDay;
104     month = startMonth;
105     year  = startYear;
106 }
107
108 //method for printing the attributes with the informations stored
109 //in them from the client to the screen
110 void employee::print()
```

```
111 {
112
113     cout << left;
114     cout << fixed <<setprecision(2);
115     cout << setw(15)  << name      << setw(9) << id      << setw(16)
116          << phoneNum  << setw(8)   << age      << setw(7) << gender
117          << setw(15)  << jobTitle  << setw(15) <<salary << month << "/"
118          << day << "/" << year << endl;
119 }
120
121 /*****************************************************************************
122  * Methods for class programmer
123  *****************************************************************************/
124
125 //Default constructor
126 programmer::programmer()
127 {
128     depNumber.clear();
129     supName.clear();
130
131     percSalaryInc = 0;
132
133     cpp  = false;
134     java = false;
135 }
136
137 //Destructor
138 programmer::~programmer() {}
139
140 //Method for getting the department number from the client and
141 //store it in depNumber number attribute
142 void programmer::setDepartmentNum(string depNum)
143 {
144     depNumber = depNum;
145 }
146
147 //Method for getting the supervisore's name from the clinet and
148 //store it in the supName attribute
149 void programmer::setSupName(string supervisor)
150 {
151     supName = supervisor;
152 }
153
154 //Method for getting the salary increase percentage from the client
155 //store it in the percSalaryInc attribute
156 void programmer::setPercentage(int percent)
157 {
158     percSalaryInc = percent;
159 }
160
161 //Method for getting the C++ identifier true/false result from the
162 //client and store it in the cpp attribute
163 void programmer::cppIdentifier(bool cppIdent)
164 {
165     cpp = cppIdent;
```

```
166
167
168 }
169
170 //Method for getting the Java identifier true/false result from the
171 //client and store it in the java identifier
172 void programmer::javaIdentifier(bool javaIdent)
173 {
174     java = javaIdent;
175
176 }
177
178 //method for printing the attributes with the informations stored
179 //in them from the client to the screen
180 void programmer::printProgrammer()
181 {
182     cout << left;
183     cout << setw(15) << name    << setw(15)      << depNumber << setw(20)
184          << supName  << "%" << setw(15) << percSalaryInc;
185
186     if(cpp == true)
187     {
188         cout << setw(22) << "Yes";
189     }
190
191     else
192     {
193         cout << setw(22) << "No";
194     }
195
196     if(java == true)
197     {
198         cout << "Yes";
199     }
200
201     else
202     {
203         cout << "No";
204     }
205
206     cout << endl;
207 }
208
209
210 /*****************************************************************************
211  * Methods for class softArch
212  *****************************************************************************/
213
214 //Default constructor
215 softArch::softArch()
216 {
217     depNumber.clear();
218     supName.clear();
219
220     percSalaryInc = 0;
```

```
221     experience    = 0;
222 }
223
224 //Destructor
225 softArch::~softArch() {}
226
227 //Method for getting the department number from the client and
228 //store it in depNumber number attribute
229 void softArch::setDepartmentNum(string departmnet)
230 {
231     depNumber = departmnet;
232 }
233
234 //Method for getting the supervisore's name from the clinet and
235 //store it in the supName attribute
236 void softArch::setSupName(string supervisor)
237 {
238     supName = supervisor;
239 }
240
241 //Method for getting the salary increase percentage from the client
242 //store it in the percSalaryInc attribute
243 void softArch::setPercentage(float percent)
244 {
245     percSalaryInc = percent;
246 }
247
248 //Method for getting the experience increase percentage from the client
249 //store it in the experience attribute
250 void softArch::setExperience(int empYears)
251 {
252     experience = empYears;
253 }
254
255 //method for printing the attributes with the informations stored
256 //in them from the client to the screen
257 void softArch::printSoftArch()
258 {
259     cout << left;
260     cout << setw(14) << name    << setw(18) << depNumber     << setw(18)
261          << supName  << "%"     << setw(16) << percSalaryInc << experience
262          << endl;
263 }
264
265
266
267
268
```

```
Data:
CS1C Employees
Name           ID       Phone #        Age    Gender   Job title      Salary         Hire date
-------------------------------------------------------------------------------------------------
Tom Brady      12345    949-555-1234    42      M      Quarterback    8000000.00     8/31/2018
Aaron Rodgers  12346    310-555-5555    36      M      Quarterback    770123.00      5/8/2019
Oprah Winfrey  98765    703-703-1234    64      F      Talk show Host 9900000.00     12/25/2017
Jay Leno       77777    203-555-6789    69      M      Comedian       500500.00      3/1/2012


Programmers:
Name           ID       Phone #        Age    Gender   Job title      Salary         Hire date
-------------------------------------------------------------------------------------------------
Sam Software   54321    819-123-4567    21      M      Programmer     223000.00      12/24/2017
Mary Coder     65432    310-555-5555    28      F      Programmer     770123.00      2/8/2019


Name          Department    Supervisor's Name     Raise %     C++ Knowledge      Java Knowledge
-------------------------------------------------------------------------------------------------
Sam Software   5432122        Joe Boss            %4.00           Yes                  No
Mary Coder     6543222        Mary Leader         %7.00           Yes                  Yes


Software Architects:
Name           ID       Phone #        Age    Gender   Job title      Salary         Hire date
-------------------------------------------------------------------------------------------------
Alex Arch      88888    819-123-4444    31      M      Architect      323000.00      12/24/2018
Sally Designer 87878    310-555-8888    38      F      Architect      870123.00      2/8/2013


Name         Department     Supervisor's Name     Raise %     Years of experience
-------------------------------------------------------------------------------------------------
Alex Arch     5434222           Big Boss          %5.00            4
Sally Designer6543422           Big Boss          %8.00            11
```