

methods.cpp

```
1 /*****
2  * PROGRAMMER : Ali Eshghi
3  * STUDENT ID : 1112261
4  * CLASS      : CS1C
5  * SECTION    : MW 5pm
6  * Assign #1   : CS1C corporation
7  * DUE DATE    : 29 January 2020
8  *****/
9
10 #include "MyHeader.h"
11
12 /*****
13  * Methods for class date
14  *****/
15
16
17 //non-Default constructor
18 date::date()
19 {
20     //INITIALIZATION
21     day   = 0;
22     month = 0;
23     year  = 0;
24 }
25
26 //destructor
27 date::~date() {}
28
29
30 /*****
31  * Methods for class employee
32  *****/
33
34 //default constructor
35 employee::employee()
36 {
37     //INITIALIZATION
38     name.clear();
39     id.clear();
40     phoneNum.clear();
41     jobTitle.clear();
42
43     age   = 0;
44     salary = 0;
45
46     gender = ' ';
47 }
48
49
50 //destructor
51 employee::~employee() {}
52
53 //method for getting the name from the client and store it in name attribute
54 void employee::setName(string empName)
55 {
```

```

56     name = empName;
57 }
58
59 //method 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 supervisor's name from the client 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 supervisor's name from the client 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

```