

MyHeader.h

```
1 /*****
2  * PROGRAMMER : Ali Eshghi
3  * STUDENT ID : 1112261
4  * CLASS      : CS1C
5  * SECTION    : MW 5pm
6  * Assign #4   : Enhanced Employee
7  * DUE DATE    : 26 February 2020
8  *****/
9 #ifndef MYHEADER_H_
10 #define MYHEADER_H_
11
12 //Preprocessor directives
13
14 #include<iostream> //for input and output
15 #include<iomanip>   //for output style
16 #include<string>    //for using string
17
18 //using the name space standard
19 using namespace std;
20
21
22 //class date: for defining the date
23 class date
24 {
25 //public parts containing the method functions of the class
26 public:
27
28     //default constructor
29     date();
30
31     //destructor
32     ~date();
33
34 //protected attributes of the class (accessible by derived classes)
35 protected:
36     int month; //PROCESS - for storing month
37     int day;   //PROCESS - for storing day
38     int year;  //PROCESS - for storing year
39 };
40
41
42 //Class employee(derived from the class date):
43 //for setting and changing the attributes about the employees
44 class employee: protected date
45 {
46 //public parts containing the method functions of the class
47 public:
48     //default constructor
49     employee();
50
51
52     //destructor
53     ~employee();
54
55     //method function for setting the name
56     void setName(string empName);
57
58     string getName();
59 }
```

```

60 //method function for setting the Id
61 void setId(string empId);
62
63 //method function for setting the phone number
64 void setPhoneNumber(string Number);
65
66 //method function for setting the age
67 void setAge(int empAge);
68
69 //method function for setting the gender
70 void setGender(char sex);
71
72 //method function for setting the job title
73 void setJobTitle(string title);
74
75 //method function for setting the salary
76 void setSalary(double income);
77
78 //method function for setting the hire date
79 void setDate(int startDay, int startMonth, int startYear);
80
81 //method function for printing the attributes of the class
82 void print();
83
84 //protected attributes of the class (accessible by derived classes)
85 protected:
86     string name;        //PROCESS - storing the name
87     string id;          //PROCESS - storing the id number
88     string phoneNum;    //PROCESS - storing the phone number
89     int age;            //PROCESS - storing the age
90     char gender;        //PROCESS - storing the gender
91     string jobTitle;    //PROCESS - storing the job title
92     double salary;      //PROCESS - storing the salary
93     date startDate;     //PROCESS - storing the hire date
94 };
95
96
97 //class softTester(derived from the class employee):
98 //for setting and changing the attributes about the programmer
99
100 class softTester: public employee
101 {
102
103 //public parts containing the method functions of the class
104 public:
105     //default constructor
106     softTester(string defAddress, string defCity, string defState, int defZipCode);
107
108     //copy constructor
109     softTester(const softTester &obj);
110
111     //destructor
112     ~softTester();
113
114     //Method for getting the address,
115     //return type: String
116     string getAddress(void);
117
118     //Method for getting the City,

```

```

119 //return type: String
120 string getCity(void);
121
122 //Method for getting the State,
123 //return type: String
124 string getState(void);
125
126 //Method for getting the zip code,
127 //return type: integer
128 int getZipCode(void);
129
130 //Method for changing the address attribute
131 //of the class softTester
132 void changeAddress(string newAddress);
133
134 //Method for changing the city attribute
135 //of the class softTester
136 void changeCity(string newCity);
137
138 //Method for changing the state attribute
139 //of the class softTester
140 void changeState(string newState);
141
142 //Method for changing the zipCode attribute
143 //of the class softTester
144 void changeZipCode(int newZipCode);
145
146 //Method for displaying the attributes
147 void softTesterDisplay();
148
149 //private part of the class; containing the attributes of the class
150 private:
151     string* address;//PROCESS - storing the department number
152     string* city;      //PROCESS - storing the supervisor's name
153     string* state;     //PROCESS - storing the salary increase percentage
154     int* zipCode;//PROCESS - storing if the person knows c++
155 };
156
157
158
159
160 #endif /* MYHEADER_H_ */
161

```