```cpp
/*****************************************************************************
 * PROGRAMMER : Ali Eshghi
 * STUDENT ID : 1112261
 * CLASS      : CS1C
 * SECTION    : MW 5pm
 * Assign #4  : Enhanced Employee
 * DUE DATE   : 26 Febuary 2020
 *****************************************************************************/

#include"MyHeader.h"

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

//non-Default constructor
date::date()
{
    //INITIALIZATION
    cout << "date class constructor called" << endl;
    day   = 0;
    month = 0;
    year  = 0;
}

//destruactor
date::~date()
{
    cout << "date class destructor called" << endl;
}


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

//default constructor
employee::employee()
{
    //INITIALIZATION
    cout << "employee constructor called" << endl;
    name.clear();
    id.clear();
    phoneNum.clear();
    jobTitle.clear();

    age    = 0;
    salary = 0;

    gender = ' ';

}

//destructor
employee::~employee()
{
    cout << "employee class destructor called" << endl;
}

```

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

string employee::getName()
{
    return name;
}

//methpod for getting the id from the client and store it in id attribute
void employee::setId(string empId)
{
    id = empId;
}

//method for getting the phone number from the client and store it in
//the phoenNum attribute
void employee::setPhoneNumber(string number)
{
    phoneNum = number;
}

//method for getting the age from the client and store it in
//age attribute
void employee::setAge(int empAge)
{
    age = empAge;
}
//method for getting the gender from the client and store it
//in gender attribute
void employee::setGender(char sex)
{
    gender = sex;
}

//method for getting the job title from the client and store
//it in the jobTile attribute
void employee::setJobTitle(string title)
{
    jobTitle = title;
}

//method for getting the salary from the client and store it
//in salary attribute
void employee::setSalary(double income)
{
    salary = income;
}

//method for getting the hire date attributes and save the date
//into the attributes of day, month, and year
void employee::setDate(int startDay, int startMonth, int startYear)
{
    day   = startDay;
    month = startMonth;
    year  = startYear;
}
```

```cpp

//method for printing the attributes with the informations stored
//in them from the client to the screen
void employee::print()
{

    cout << left;
    cout << fixed <<setprecision(2);
    cout << setw(15)  << name      << setw(9) << id      << setw(16)
         << phoneNum  << setw(8)   << age      << setw(7) << gender
         << setw(15)  << jobTitle  << "$" << setw(15) << salary << month << "/"
         << day << "/" << year << endl;
}



/*****************************************************************************
 * Methods for class softTester
 *****************************************************************************/

//Default cosntructor
softTester::softTester(string defAddress, string defCity, string defState, int
   defZipCode)
{
    cout << "\nNormal softTester class constructor allocating ptr." << endl;

    //allocate memory for the pointer

    address = new string;
    *address = defAddress;

    city = new string;
    *city = defCity;

    state = new string;
    *state = defState;

    zipCode = new int;
    *zipCode = defZipCode;
}


//copy constructor
softTester::softTester(const softTester& obj)
{
    //using the deep copying to copy
    cout << "\nCopy softtester class constructor allocating the ptr. " << endl;

    //first we need to deallocate any value that this string is holding
    delete address;
    delete city;
    delete state;
    delete zipCode;

    //address is a pointer, so we need to deep copy it if it is non-null
    if(obj.address)
    {
        //allocate memory for our copy
        address = new string;
```

```
177         city    = new string;
178         state   = new string;
179         zipCode = new int;
180 
181         //do the copy
182         address = obj.address;
183         city    = obj.city;
184         state   = obj.state;
185         zipCode = obj.zipCode;
186     }
187 
188     else
189     {
190         address = NULL;
191         city    = NULL;
192         state   = NULL;
193         zipCode = NULL;
194     }
195 }
196 
197 //destructor
198 softTester::~softTester(void)
199 {
200     cout << "SoftTester class destructor called " << endl << endl;
201     delete address;
202     delete city;
203     delete state;
204     delete zipCode;
205 }
206 
207 //Method for getting the address,
208 //return type: String
209 string softTester::getAddress(void)
210 {
211     return *address;
212 }
213 
214 //Method for getting the City,
215 //return type: String
216 string softTester::getCity(void)
217 {
218     return *city;
219 }
220 
221 //Method for getting the State,
222 //return type: String
223 string softTester::getState(void)
224 {
225     return *state;
226 }
227 
228 //Method for getting the zip code,
229 //return type: integer
230 int softTester::getZipCode(void)
231 {
232     return *zipCode;
233 }
234 
235 //Method for changing the address attribute
```

```cpp
//of the class softTester
void softTester::changeAddress(string newAddress)
{
    address = new string;
    *address = newAddress;

}

//Method for changing the city attribute
//of the class softTester
void softTester::changeCity(string newCity)
{
    city = new string;
    *city = newCity;
}

//Method for changing the state attribute
//of the class softTester
void softTester::changeState(string newState)
{
    state = new string;
    *state = newState;
}

//Method for changing the zipCode attribute
//of the class softTester
void softTester::changeZipCode(int newZipCode)
{
    zipCode = new int;
    *zipCode = newZipCode;
}

//Method for diplayong the attributes
void softTester::softTesterDisplay()
{
    cout << left;
    cout << fixed <<setprecision(2);
    cout << setw(15)   << employee::getName()
         << setw(23)   << *address  << setw(16)
         << *city      << setw(8)   << *state    << setw(7) << *zipCode
         << endl;

}
```