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

//Preprocessor directives

#include<iostream> //for input and output
#include<iomanip>  //for output style
#include<string>   //for using string

//using the name space standard
using namespace std;


//class date: for defining the date
class date
{
//public parts containing the method functions of the class
public:

    //default constructor
    date();

    //destructor
    ~date();

//protected attributes of the class (accessible by derived classes)
protected:
    int month;  //PROCESS - for storing month
    int day;//PROCESS - for storing day
    int year;   //PROCESS - for storing year
};


//Class employee(derived from the class date):
//for setting and changing the attributes about the employees
class employee: protected date
{
//public parts containing the method functions of the class
public:
    //default constructor
    employee();


    //destructor
    ~employee();

    //method function for setting the name
    void setName(string empName);

    string getName();

```

```cpp
/*************************************************************************
 * PROGRAMMER : Ali Eshghi
 * STUDENT ID : 1112261
 * CLASS      : CS1C
 * SECTION    : MW 5pm
 * Assign #4  : Enhanced Employee
 * DUE DATE   : 26 Febuary 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
    softTester  softTester1(" ", " ", " ", 0); //PROCESS - softTester
                                               // class type variable
    softTester  softTester2 = softTester1;  //PROCESS - calling the copy constructor


    cout << "Software testers:" << endl << 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;


    //passing the employees information
    //to the methods to set the data

    employee.setName("Joe Calculus");
    employee.setId("64879");
    employee.setPhoneNumber("949-555-1234");
    employee.setAge(42);
    employee.setGender('M');
    employee.setJobTitle("Math Wiz");
    employee.setSalary(110000);
    employee.setDate(31,8,2017);

    //printing the information
    employee.print();

    //passing the employees information
    //to the methods to set the data

    employee.setName("Mary Algebra");
    employee.setId("76309");
    employee.setPhoneNumber("213-555-5555");
    employee.setAge(22);
    employee.setGender('F');
    employee.setJobTitle("Math Helper");
    employee.setSalary(170123);
    employee.setDate(8,5,2017);

    //printing the information
    employee.print();

    //passing the employees information
    //to the methods to set the data

    employee.setName("Joe Trig");
    employee.setId("10192");
    employee.setPhoneNumber("714-703-1234");
    employee.setAge(29);
    employee.setGender('F');
    employee.setJobTitle("Math Contact");
    employee.setSalary(200000);
    employee.setDate(25,12,2016);

    //printing the information
    employee.print();


    cout << endl << endl;
    cout << left;
    cout << setw(15)  << "Name"      << setw(23) << "Address"    << setw(15)
         << "City" << setw(7)     << "State"    << setw(9) << "Zip Code"
         << endl;
    cout << "-------------------------------------------------";
    cout << "-------------------------------------------------";
    cout << endl;

    //passing the software Tester information
    //to the employees methods to set the data

    softTester1.setName("Joe calCules");
    softTester1.changeAddress("1234 Main Avenue");
    softTester1.changeCity("Laguna Niguel");
    softTester1.changeState("CA");
    softTester1.changeZipCode(92677);

    //printing the employee informations of the software tester
    softTester1.softTesterDisplay();

    //passing the software Tester information
    //to the employees methods to set the data

    softTester1.setName("Mary Algebra");
    softTester1.changeAddress("3333 Marguerite Pkwy");
    softTester1.changeCity("Mission Viejo");
    softTester1.changeState("CA");
    softTester1.changeZipCode(92646);

    //printing the employee informations of the software tester
    softTester1.softTesterDisplay();

    //passing the software Tester information
    //to the employees methods to set the data

    softTester1.setName("jo Trig");
    softTester1.changeAddress("9876 Elm Street");
    softTester1.changeCity("San Clemente");
    softTester1.changeState("CA");
    softTester1.changeZipCode(92672);

    //printing the employee informations of the software tester
    softTester1.softTesterDisplay();

    return 0;
}


```

```
/*************************************************************************
 * 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
    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)
{
    name = empName;
}

string employee::getName()
{
```

```cpp
    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;
}

//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 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 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;
        city    = new string;
        state   = new string;
        zipCode = new int;

        //do the copy
        address = obj.address;
        city    = obj.city;
        state   = obj.state;
        zipCode = obj.zipCode;
    }

    else
    {
        address = NULL;
        city    = NULL;
        state   = NULL;
        zipCode = NULL;
    }
}

//destructor
softTester::~softTester(void)
{
    cout << "\n\nfreeing memory" << endl << endl;
    delete address;
    delete city;
    delete state;
    delete zipCode;
}

//Method for getting the address,
//return type: String
string softTester::getAddress(void)
{
    return *address;
}

//Method for getting the City,
//return type: String
string softTester::getCity(void)
{
    return *city;
}

//Method for getting the State,
//return type: String
string softTester::getState(void)
{
    return *state;
}

//Method for getting the zip code,
//return type: integer
int softTester::getZipCode(void)
{
    return *zipCode;
}

//Method for changing the address attribute
//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;

}


```