```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)
{
    name = empName;
}

//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 programmer
 *******************************************************************************/

//Default constructor
programmer::programmer()
{
    depNumber.clear();
    supName.clear();

    percSalaryInc = 0;

    cpp  = false;
    java = false;
}

//Destructor
programmer::~programmer() {}

//Method for getting the department number from the client and
//store it in depNumber number attribute
void programmer::setDepartmentNum(string depNum)
{
    depNumber = depNum;
}

//Method for getting the supervisore's name from the clinet and
//store it in the supName attribute
void programmer::setSupName(string supervisor)
{
    supName = supervisor;
}

//Method for getting the salary increase percentage from the client
//store it in the percSalaryInc attribute
void programmer::setPercentage(int percent)
{
    percSalaryInc = percent;
}

//Method for getting the C++ identifier true/false result from the
//client and store it in the cpp attribute
void programmer::cppIdentifier(bool cppIdent)
{
    cpp = cppIdent;
```

```


}

//Method for getting the Java identifier true/false result from the
//client and store it in the java identifier
void programmer::javaIdentifier(bool javaIdent)
{
    java = javaIdent;

}

//method for printing the attributes with the informations stored
//in them from the client to the screen
void programmer::printProgrammer()
{
    cout << left;
    cout << setw(15) << name    << setw(15)      << depNumber << setw(20)
         << supName  << "%" << setw(15) << percSalaryInc;

    if(cpp == true)
    {
        cout << setw(22) << "Yes";
    }

    else
    {
        cout << setw(22) << "No";
    }

    if(java == true)
    {
        cout << "Yes";
    }

    else
    {
        cout << "No";
    }

    cout << endl;
}


/*******************************************************************************
 * Methods for class softArch
 *******************************************************************************/

//Default constructor
softArch::softArch()
{
    depNumber.clear();
    supName.clear();

    percSalaryInc = 0;
```

```cpp
    experience    = 0;
}

//Destructor
softArch::~softArch() {}

//Method for getting the department number from the client and
//store it in depNumber number attribute
void softArch::setDepartmentNum(string departmnet)
{
    depNumber = departmnet;
}

//Method for getting the supervisore's name from the clinet and
//store it in the supName attribute
void softArch::setSupName(string supervisor)
{
    supName = supervisor;
}

//Method for getting the salary increase percentage from the client
//store it in the percSalaryInc attribute
void softArch::setPercentage(float percent)
{
    percSalaryInc = percent;
}

//Method for getting the experience increase percentage from the client
//store it in the experience attribute
void softArch::setExperience(int empYears)
{
    experience = empYears;
}

//method for printing the attributes with the informations stored
//in them from the client to the screen
void softArch::printSoftArch()
{
    cout << left;
    cout << setw(14) << name    << setw(18) << depNumber     << setw(18)
         << supName  << "%"     << setw(16) << percSalaryInc << experience
         << endl;
}
```