```
//Francisco Fierro
//Lab 5
//Employee.h

#ifndef H_EMPLOYEE
#define H_EMPLOYEE

#include "Date.h"

using namespace std;

class Employee{
public:
	virtual void putData();
	void setFirstName(string n);
	string getFirstName() const;
	void setLastName(string n);
	string getLastName() const;
	void setID(string i);
	string getID() const;
	void setBDay(int m, int d, int y);
	void setBDay(Date b);
	void setSex(char s);
	char getSex() const;
	virtual float monthlyEarning() = 0;
	Employee(string n1 = "Jane", string n2 = "Doe", string i = "0",
		Date b = Date(), char s = 'F');

protected:
	string firstName;
	string lastName;
	string id;
	char sex;
	Date bDay;
};
#endif


//Francisco Fierro
//Lab 5
//Employee.cpp

#include <iostream>
#include <iomanip>
#include "Employee.h"

using namespace std;

void Employee::putData(){
```

```
        cout << "Employee ID Number: " << id << endl;
        cout << "Employee Name: " << firstName << " " << lastName << endl;
        cout << "Birth Date: ";
        bDay.showDate();
}

void Employee::setFirstName(string n){
        firstName = n;
}

string Employee::getFirstName() const{
        return firstName;
}

void  Employee::setLastName(string n){
        lastName = n;
}

string Employee::getLastName() const{
        return lastName;
}

void Employee::setID(string i){
        id = i;
}

string Employee::getID() const{
        return id;
}

void Employee::setBDay(int m, int d, int y){
        bDay = Date(m,d,y);
}

void Employee::setBDay(Date b){
        bDay = b;
}

void Employee::setSex(char s){
        sex = s;
}

char Employee::getSex() const{
        return sex;
}

Employee::Employee(string n1, string n2, string i, Date b, char s):
        firstName(n1), lastName(n2), id(i), bDay(b), sex(s) {}
```

```
//Francisco Fierro
//Lab 5
//Faculty.h

#ifndef H_FACULTY
#define H_FACULTY

#include <iostream>
#include "Employee.h"
#include "Education.h"
using namespace std;

class Faculty: public Employee{
public:
	void putData();
	void setLevel(string l);
	string getLevel() const;
	void setEd(Education e);
	Education getEd() const;
	float monthlyEarning();
	Faculty(string fn = "Jane" , string ln  = "Doe" , string i = 0 ,
		Date b = Date(), char s = 'F'   , string l = "MS",
		Education e = Education());

protected:
	string level;
	Education ed;
};

#endif

//Francisco Fierro
//Lab 5
//Faculty.cpp

#include "Faculty.h"
#include "constants.h"
#include "Education.h"
#include <iostream>
#include <iomanip>

using namespace std;

void Faculty::putData(){
	Employee::putData();
	string title = "";
	if(level.compare("AS") == 0){
		title = "Assistant";
	}
```

```
	else if(level.compare("AO") == 0){
		title = "Associate";
	}
	else if(level.compare("FU") == 0){
		title = "Full";
	}
	cout << title << " Professor " << firstName << " " << lastName
		<< endl;
	cout << "Montly Salary: " << fixed << showpoint <<
		setprecision(2) << monthlyEarning() << endl;
}

void Faculty::setLevel(string l){
	level = l;
}

string Faculty::getLevel() const{
	return level;
}

void Faculty::setEd(Education e){
	ed = e;
}

Education Faculty::getEd() const{
	return ed;
}

float Faculty::monthlyEarning(){
	if(level.compare("AS") == 0){
		return FACULTY_MONTHLY_SALARY;
	}
	else if(level.compare("AO") == 0){
		return FACULTY_MONTHLY_SALARY * 1.2;
	}
	else if(level.compare("FU") == 0){
		return FACULTY_MONTHLY_SALARY * 1.4;
	}
}

Faculty::Faculty(string fn, string ln, string i, Date b,
	 char s, string l, Education e): Employee::Employee(
		fn, ln, i, b, s), level(l), ed(e){}



//Francisco Fierro
//Lab 5
//testMain.cpp
```

```
//Main file for Lab 5

#include <iostream>
#include <iomanip>
#include "Staff.h"
#include "Faculty.h"
#include "Parttime.h"
#include <typeinfo>

int main(){

	//create the Employees
	Employee *empPtr[9];

	Date bDay = Date(2,23,59);
	empPtr[0] = new Staff("Paita", "Allen", "123",
			bDay, 'M', 50.00);

	bDay.setDate(7,12,64);
	empPtr[1] = new Staff("Steven", "Zapata", "456",
		bDay, 'F', 35.00);

	bDay.setDate(6,2,70);
	empPtr[2] = new Staff("Enrique", "Rios", "789",
		bDay, 'M', 40.00);

	bDay.setDate(4,27,62);
	Education ed = Education("PhD", "Masters");
	empPtr[3] = new Faculty("Anne", "Johnson", "243",
		bDay, 'F', "FU", ed);

	bDay.setDate(3,14,75);
	ed.setDegree("PhD");
	ed.setMajor("English");
	empPtr[4] = new Faculty("William", "Bouris", "791",
		bDay, 'F', "AO", ed);

	bDay.setDate(5,22,80);
	ed.setDegree("MS");
	ed.setMajor("Physical Education");
	empPtr[5] = new Faculty("Christopher", "Andrade", "623",
		bDay, 'F', "AS", ed);

	bDay.setDate(8,10,77);
	empPtr[6] = new Parttime("Augusto", "Guzman", "455",
		bDay, 'F', 35.00, 30.0);

	bDay.setDate(9,15,87);
	empPtr[7] = new Parttime("Martin", "Depirro", "678",
```

```
		bDay, 'F', 30.00, 15.00);

	bDay.setDate(11,24,88);
	empPtr[8] = new Parttime("Marque", "Aldaco", "945",
		bDay, 'M', 35.0, 20.00 );


	//display their data
	float totalParttimeSalary = 0;
	float totalEmployeeSalary = 0;
	for(int i = 0; i < 9; i++){
		//Buch of old code
		//made putData() a virtual function
		//if(typeid(*empPtr[i]) == typeid(Staff)){
			//dynamic_cast<Staff*>(empPtr[i])->putData();
			//totalEmployeeSalary += dynamic_cast<Staff*>(empPtr[i])->monthlyEarning();


		//}
		//else if(dynamic_cast<Faculty*>(empPtr[i])){
			//dynamic_cast<Faculty*>(empPtr[i])->putData();
			//totalEmployeeSalary += dynamic_cast<Faculty*>(empPtr[i])-
>monthlyEarning();
		//}
		if(dynamic_cast<Parttime*>(empPtr[i])){
			//dynamic_cast<Parttime*>(empPtr[i])->putData();
			//totalParttimeSalary += dynamic_cast<Parttime*>(empPtr[i])-
>monthlyEarning();
			//totalEmployeeSalary += dynamic_cast<Parttime*>(empPtr[i])-
>monthlyEarning();
			totalParttimeSalary += empPtr[i]->monthlyEarning();
		}

		totalEmployeeSalary += empPtr[i]->monthlyEarning();
		empPtr[i]->putData();
		cout << "\n";
	}

		cout << "\n";
		cout << "Total Parttime Monthly Salary: " << setprecision(2) << showpoint <<
			fixed << totalParttimeSalary << endl;
		cout << "Total Employee Monthly Salary: " << fixed << totalEmployeeSalary <<
			endl;
}
```