```cpp
//
//  main.cpp
//  AbsoluteCpp_ch14_1
//

#include <iostream>
#include "hourlyemployee.h"
#include "salariedemployee.h"
using std::cout;
using std::endl;
using SavitchEmployees::HourlyEmployee;
using SavitchEmployees::SalariedEmployee;

int main( )
{
    HourlyEmployee joe;
    joe.setName("Mighty Joe");
    joe.setSsn("123-45-6789");
    joe.setRate(20.50);
    joe.setHours(40);
    cout << "Check for " << joe.getName( )
         << " for " << joe.getHours( ) << " hours.\n";
    joe.printCheck( );
    cout << endl;

    SalariedEmployee boss("Mr. Big Shot", "987-65-4321", 10500.50);
    cout << "Check for " << boss.getName( ) << endl;
    boss.printCheck( );

    return 0;
}
```

```
//This is the header file employee.h.
//This is the interface for the class Employee.
//This is primarily intended to be used as a base class to derive
//classes for different kinds of employees.
#ifndef EMPLOYEE_H
#define EMPLOYEE_H

#include <string>
using std::string;

namespace SavitchEmployees
{

    class Employee
    {
    public:
        Employee( );
        Employee(const string& theName, const string& theSsn);
        string getName( ) const;
        string getSsn( ) const;
        double getNetPay( ) const;
        void setName(const string& newName);
        void setSsn(const string& newSsn);
        void setNetPay(double newNetPay);
        void printCheck( ) const;
    private:
        string name;
        string ssn;
        double netPay;
    };

}//SavitchEmployees

#endif //EMPLOYEE_H
```

```cpp
//This is the file: employee.cpp
//This is the implementation for the class Employee.
//The interface for the class Employee is in the header file employee.h.
#include <string>
#include <cstdlib>
#include <iostream>
#include "employee.h"
using std::string;
using std::cout;

namespace SavitchEmployees {
Employee::Employee() :
        name("No name yet"), ssn("No number yet"), netPay(0) {
    //deliberately empty
}

Employee::Employee(const string& theName, const string& theNumber) :
        name(theName), ssn(theNumber), netPay(0) {
    //deliberately empty
}

string Employee::getName() const {
    return name;
}

string Employee::getSsn() const {
    return ssn;
}

double Employee::getNetPay() const {
    return netPay;
}

void Employee::setName(const string& newName) {
    name = newName;
}

void Employee::setSsn(const string& newSsn) {
    ssn = newSsn;
}

void Employee::setNetPay(double newNetPay) {
    netPay = newNetPay;
}

void Employee::printCheck() const {
    cout << "\nERROR: printCheck FUNCTION CALLED FOR AN \n"
            << "UNDIFFERENTIATED EMPLOYEE. Aborting the program.\n"
            << "Check with the author of the program about this bug.\n";
    exit(1);
}

} //SavitchEmployees
```

```
//This is the header file hourlyemployee.h.
//This is the interface for the class HourlyEmployee.
#ifndef HOURLYEMPLOYEE_H
#define HOURLYEMPLOYEE_H

#include <string>
#include "employee.h"

using std::string;

namespace SavitchEmployees {

class HourlyEmployee: public Employee {
public:
    HourlyEmployee();
    HourlyEmployee(const string& theName, const string& theSsn,
            double theWageRate, double theHours);
    void setRate(double newWageRate);
    double getRate() const;
    void setHours(double hoursWorked);
    double getHours() const;
    void printCheck();
private:
    double wageRate;
    double hours;
};

} //SavitchEmployees

#endif //HOURLYMPLOYEE_H
```

```cpp
//This is the file: hourlyemployee.cpp
//This is the implementation for the class HourlyEmployee.
//The interface for the class HourlyEmployee is in
//the header file hourlyemployee.h.
#include <string>
#include <iostream>
#include "hourlyemployee.h"
using std::string;
using std::cout;
using std::endl;

namespace SavitchEmployees {

HourlyEmployee::HourlyEmployee() :
        Employee(), wageRate(0), hours(0) {
    //deliberately empty
}

HourlyEmployee::HourlyEmployee(const string& theName, const string&
 theNumber,
        double theWageRate, double theHours) :
        Employee(theName, theNumber), wageRate(theWageRate),
         hours(theHours) {
    //deliberately empty
}

void HourlyEmployee::setRate(double newWageRate) {
    wageRate = newWageRate;
}

double HourlyEmployee::getRate() const {
    return wageRate;
}

void HourlyEmployee::setHours(double hoursWorked) {
    hours = hoursWorked;
}

double HourlyEmployee::getHours() const {
    return hours;
}

void HourlyEmployee::printCheck() {
    setNetPay(hours * wageRate);
    cout << "\n_____\n";
    cout << "Pay to the order of " << getName() << endl;
    cout << "The sum of " << getNetPay() << " Dollars\n";
    cout << "_____\n";
    cout << "Check Stub: NOT NEGOTIABLE\n";
    cout << "Employee Number: " << getSsn() << endl;
    cout << "Hourly Employee. \nHours worked: " << hours << " Rate: "
            << wageRate << " Pay: " << getNetPay() << endl;
    cout << "_____\n";
}
} //SavitchEmployees
```

```
//This is the header file salariedemployee.h.
//This is the interface for the class SalariedEmployee.
#ifndef SALARIEDEMPLOYEE_H
#define SALARIEDEMPLOYEE_H

#include <string>
#include "employee.h"

using std::string;

namespace SavitchEmployees
{

    class SalariedEmployee : public Employee
    {
    public:
        SalariedEmployee( );
        SalariedEmployee (const string&  theName, const string&  theSsn,
                                  double theWeeklySalary);
        double getSalary( ) const;
        void setSalary(double newSalary);
        void printCheck( );
    private:
        double salary;//weekly
    };

}//SavitchEmployees

#endif //SALARIEDEMPLOYEE_H
```

```
//This is the file salariedemployee.cpp
//This is the implementation for the class SalariedEmployee.
//The interface for the class SalariedEmployee is in
//the header file salariedemployee.h.
#include <iostream>
#include <string>
#include "salariedemployee.h"
using std::string;
using std::cout;
using std::endl;

namespace SavitchEmployees
{
    SalariedEmployee::SalariedEmployee( ) : Employee( ), salary(0)
    {
        //deliberately empty
    }

    SalariedEmployee::SalariedEmployee(const string&  newName, const
     string&  newNumber,
                                       double newWeeklyPay)
                    : Employee(newName, newNumber), salary(newWeeklyPay)
    {
        //deliberately empty
    }

    double SalariedEmployee::getSalary( ) const
    {
        return salary;
    }

    void SalariedEmployee::setSalary(double newSalary)
    {
        salary = newSalary;
    }

    void SalariedEmployee::printCheck( )
    {
        setNetPay(salary);
        cout << "\n_____\n";
        cout << "Pay to the order of " << getName( ) << endl;
        cout << "The sum of " << getNetPay( ) << " Dollars\n";
        cout << "_____\n";
        cout << "Check Stub NOT NEGOTIABLE \n";
        cout << "Employee Number: " << getSsn( ) << endl;
        cout << "Salaried Employee. Regular Pay: "
             << salary << endl;
        cout << "_____\n";
    }
}//SavitchEmployees
```