CSC 270 – Survey of Programming Languages

C++ Lecture 5 – Inheritance

Employee.h

```
#ifndef     EMPLOYEE_H
#define     EMPLOYEE_H
#include    <string>

using namespace std;

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

#endif
```

Employee.cpp

```
#include    <cstring>
#include    <cstdlib>
#include    <iostream>
#include "Employee.h"

using namespace std;
```

```
Employee::Employee() : name ("No name yet"),
                   ssn ("No number yet"), netPay(0)
{
      // Deliberately empty

      /*
       * equivalent to
       * name = "no name yet";
       * ssn = "No number yet";
       * netPay = 0;
       */
}
```

```
Employee::Employee(string theName, string theSsn) :
name (theName), ssn (theSsn), 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(string newName)
{
      name = newName;
}
```

```
void  Employee::setSsn(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 theis progam.\n"
            << "Check with the author of the"
            << " program about this bug." << endl;

      exit(1);
}
```

HourlyEmployee.h

```
#ifndef     HOURLYEMPLOYEE_H
#define     HOURLYEMPLOYEE_H
#include    <string>

#include "Employee.h"
```

```
class HourlyEmployee : public Employee
{
public:
      HourlyEmployee(void);
      HourlyEmployee(string theName, 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;
};
#endif      //HOURLYEMPLOYEE_H
```

HourlyEmployee.cpp

```
#include    <string>
#include    <iostream>
#include "HourlyEmployee.h"

using namespace std;

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

```
HourlyEmployee::HourlyEmployee(string theName,
                  String theSsn,double theWageRate,
                              double theHours)
      : Employee(theName, theSsn),
             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" << endl;
      cout << "\n----------------------------------"
            <<    "---------------------------\n";
      cout << "Check stub: NOT NEGOTIABLE" << endl;
      cout << "Employee Number: " << getSsn()
            << endl;
```

```
      cout << " Hourly Employee.\nHours worked: "
            << hours
            << " Rate: " << wageRate << " Pay: "
            << getNetPay() << endl;
      cout << "\n----------------------------------"
            <<    "---------------------------\n";
}
```

SalariedEmployee.h

```
#ifndef     SALARIEDEMPLOYEE_H
#define     SALARIEDEMPLOYEE_H
#include    <string>
#include    "Employee.h"

using namespace std;
```

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

};
#endif      SALARIEDEMPLOYEE_H
```

SalariedEmployee.cpp

```
// This is the file hoursalariedemployee.coo
// This is the implementation for the class
SalariedEmployee
// The interface for the class SalariedEmployee is
in
// the header salariedemployee.h

#include    <iostream>
#include "SalariedEmployee.h"

SalariedEmployee::SalariedEmployee(void) : Employee(
), salary(0)
{
      // deliberately empty
}
```

```
SalariedEmployee::SalariedEmployee(string theName,
            string theNumber, double theWeeklyPay) :
Employee (theName, theNumber), salary(theWeeklyPay)
{
      // deliberately empty
}

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

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

```
void  SalariedEmployee::printCheck(void)
{
      setNetPay(salary);

      cout << "\n-----------------------------------"
            << "---------------------------\n";
      cout << "Pay to the order of " << getName()
            << endl;
      cout << "The sum of " << getNetPay()
            << " Dollars" << endl;
      cout << "\n-----------------------------------"
            << "---------------------------\n";
      cout << "Check stub: NOT NEGOTIABLE" << endl;
      cout << "Employee Number: " << getSsn()
            << endl;
```

```
      cout << " Salaried Employee.  Regular Pay: "
            << salary << endl;
      cout << "\n-----------------------------------"
            << "---------------------------\n";
```

`Employee.h` with **`protected`** Properties

```
#ifndef     EMPLOYEE_H
#define     EMPLOYEE_H
#include    <string>

using namespace std;

class Employee    {
public:
      Employee();
      Employee(string theName, string theSsn);
      string      getName() const;
      string      getSsn() const;
      double      getNetPay() const;
      void  setName(string newName);
```

```
      void  setSsn(string newSsn);
      void  setNetPay(double newNetPay);
      void  printCheck() const;
protected:
      string name;
      string ssn;
      double netPay;
};

#endif
```

Using **protected** Properties

```
void  HourlyEmployee::printCheck()
{
      setNetPay(hours * wageRate);
      cout << "\n----------------------------------"
            <<    "---------------------------\n";
      cout << "Pay to the order of " << name
            << endl;
      cout << "The sum of " << netPay
             << " Dollars" << endl;
      cout << "\n----------------------------------"
            <<    "---------------------------\n";
      cout << "Check stub: NOT NEGOTIABLE" << endl;
      cout << "Employee Number: " << ssn << endl;
```

```
      cout << " Hourly Employee.\nHours worked: "
            << hours
            << " Rate: " << wageRate << " Pay: "
            << netPay << endl;
      cout << "\n----------------------------------"
            <<    "---------------------------\n";
}
```