Homework #4

```
Print Employee 1,2,3 Salary Information:

Employee ID Number: 1
Name: Jones , Booker, T
Department Code: 22
Employee Salary: 14000

Employee ID Number: 2
Name: Hendrix , Jimi, NMI
Department Code: 14
Employee Salary: 32000

Employee ID Number: 3
Name: Morrison , Jim, D
Department Code: 3
Employee Salary: 28000

Print Employee 4,5,6 Hourly Information:

Employee ID Number: 4
Name: Richardson, Samantha M.
Department Code: 11
Total Hours Worked : 34
Hourly Rate : 12
Employee Weekly Payment : 408

Employee ID Number: 5
Name: Smith, Laura K.
Department Code: 67
Total Hours Worked : 40
Hourly Rate : 14
Employee Weekly Payment : 560
```

ECE 330 – Inheritance in Salaried and Hourly Employee Classes Report

Spring 2021

Author: Clarizza Morales

Date: March 5th, 2021

Instructor: Amir Raeisi

Course: Software Design

Table of Contents

List of Figures

Employee.cpp (Provided Base Class)

```
#include <iostream>
#include <iomanip>


#include <string>


#include "employee.hpp"

using namespace std;

// constructor

Employee::Employee(long id, const string &last, const string &first, const string &initial,
                   int dept)
{
  myIdNum = id;
  myLastName = last;
  myFirstName = first;
  myMiddleInitial = initial;
  myDeptCode = dept;
}

// Accessor function defintions

void Employee::setIdNum (const long  id)
{
  myIdNum = id;
}

long Employee:: getIdNum () const                              // get id number
{
  return myIdNum;
}


void Employee:: setLastName (const string &last)        // set last name
{
  myLastName = last;
}
```

Figure 1. Employee.cpp Base class

```
string Employee:: getLastName () const                    // return last name
{
  return myLastName;
}

void Employee:: setFirstName (const string &first)       // set first name
{
  myFirstName = first;
}


string Employee:: getFirstName () const                  // return first name
{
  return myFirstName;
}


void Employee:: setMiddleInitial (const string &last)    // set middle initial
{
  myMiddleInitial = last;
}


string Employee:: getMiddleInitial () const              // return middle initial
{
  return myMiddleInitial;
}


void Employee::setDeptCode (const int dc)                // set department code
{
  myDeptCode = dc;
}


int Employee:: getDeptCode () const                              // get department code
{
  return myDeptCode;
}
```

Figure 2. Employee.cpp Base class

```
int Employee:: getDeptCode () const                              // get department code
{
  return myDeptCode;
}


void Employee::printEmployee()                    // print Employee information
{
  cout << endl;
  cout << "Employee ID Number: " << getIdNum() << endl;
  cout << "Name: " << getLastName() <<", " << getFirstName() << " " <<
               getMiddleInitial() <<"." << endl;
  cout << "Dept Code: " << getDeptCode () << endl;
}
```

Figure 3. Employee.cpp Base class

Employee.hpp (Provided Base Class – Header File)

```
/* employee.h   */

#ifndef EMPLOYEE
#define EMPLOYEE

#include <string>

using namespace std;

class Employee
{
public:
  Employee(long = 0, const string & ="" , const string & ="", const string & = "", int =0);
// constructor

  void setIdNum (const long );                  // set id number
  long getIdNum () const;                              // get id number
  void setLastName (const string &);    // set last name
  string getLastName () const;                  // return last name
  void setFirstName (const string &);   // set first name
  string getFirstName () const;                 // return first name
  void setMiddleInitial (const string &);       // set middle initial
  string getMiddleInitial () const;             // return set middle initial
  void setDeptCode(const int);                  // set department code
  int getDeptCode () const;                            // get department code
  void printEmployee ();                               // print Employee information



public:
  long  myIdNum;                        //Employee id number
  string myLastName;            //Employee last name
  string myFirstName;           //Employee first name
  string myMiddleInitial;       //Employee middle intial
  int myDeptCode;                       //Department code
};

#endif
```

Figure 4. Employee.hpp Header File

salariedEmployee.cpp File

```
/* File: salariedEmployee.cpp */

#include "salariedEmployee.hpp"
//#include "employee.hpp"
#include <iomanip>
//#include <stdlib.h>
#include <iostream>
#include <string>


//Parametrized Constructor

/*
int _monthlySalary;
long _IDnum;
int _departmentCode;
std::string _FirstName;
std::string _LastName;
std::string _Minitial;
*/

SalariedEmployee::SalariedEmployee(long id, const std::string &lastName, const std::string &firstName, const std::string &initial, int depCode, int salary)
{
        monthlySalary = salary;
        myIdNum = id;
        myDeptCode = depCode;
        myFirstName = firstName;
        myLastName = lastName;
        myMiddleInitial= initial;
}

//setter for the salary of employee

void SalariedEmployee::setSalary(int salary)
{
        monthlySalary = salary;
}

//getter for salary of employee
```

Figure 5. salariedEmployee.cpp Class File

```cpp
//getter for salary of employee

int SalariedEmployee::getSalary()
{
        return monthlySalary;
}

// salary calculation function
double SalariedEmployee::calculateSalary()
{
        double fractionOftime = 1.0;
        monthlySalary = monthlySalary*fractionOftime;
        return monthlySalary;
}

//print function
void SalariedEmployee::printFunctionforSalariedEmpl()
{
        //use the methods from employee.hpp and employee.cpp
        // e.g. from test.cpp: 001, "Jones", "Booker", "T", 22
        std::cout << "\n";
        std::cout << "Employee ID Number: " << Employee::getIdNum() << "\n";
        std::cout << "Name: " << Employee::getLastName() << " , " << Employee::getFirstName() << ", " << getMiddleInitial() << "\n";
        std::cout << "Department Code: " << Employee::getDeptCode() << "\n";
        std::cout << "Employee Salary: " << SalariedEmployee::calculateSalary() << "\n";
}
```

Figure 6. salariedEmployee.cpp Class File

salariedEmployee.hpp Header File

```
/* File: salariedEmployee.hpp */

#include "employee.hpp"
#//include <iostream>
#include <string>


class SalariedEmployee:public Employee{

        /*
        int monthlySalary;
        long _IDnum;
        int _departmentCode;
        std::string _FirstName;
        std::string _LastName;
        std::string _Minitial;
*/
public:

        SalariedEmployee(long = 0, const string & ="" , const string & ="", const string & = "", int =0, int =0);
        int monthlySalary;
        void setSalary(int salary);
        int getSalary();
        double calculateSalary();
        void printFunctionforSalariedEmpl();

};
```

Figure 7. salariedEmployee.hpp Header File

hourlyEmployee.cpp File

```
/* File: hourlyEmployee.cpp */

#include "hourlyEmployee.hpp"
#include <iostream>
#include <iomanip>
#include <string>

HourlyEmployee::HourlyEmployee(long id, const std::string &lastName, const std::string &firstName, const std::string &initial,int deptCode, int totalHrs, int hourRate)
{
        totalHours = totalHrs;
        hourlyRate = hourRate;
        WeeklyPayment = 0;
        overtime = 0;
        myLastName = lastName;
        myFirstName = firstName;
        myMiddleInitial = initial;
        myDeptCode = deptCode;
        myIdNum = id;

}

void HourlyEmployee::setTotalHoursWorked(int totalHrs)
{
        totalHours = totalHrs;
}
int HourlyEmployee::getTotalHoursWorked()
{
        return totalHours;
}

void HourlyEmployee::setHourlyRate(int hourRate)
{
        hourlyRate = hourRate;
}
int HourlyEmployee::getHourlyRate()
{
        return hourlyRate;
}
```

Figure 8. hourlyEmployee.hpp Class File

```
double HourlyEmployee::calculatedHourlyPayment()
{
        //e.g toal hrs = 50, then 50 > 40, 50 - 40 = 10 overtime hours, (normalRate*overtime*1.5 + 40*normalRate)
        if ( totalHours > 40){
                overtime = totalHours-40;
                WeeklyPayment = ((hourlyRate*40) + (hourlyRate*overtime*1.5));
                return WeeklyPayment;
        }
        else{
                // e.g total hrs = 15 , 15*normalRate = totalPayment
                WeeklyPayment = (hourlyRate*totalHours);
                return WeeklyPayment;
        }

}

void HourlyEmployee::printFunctionforHourlyEmpl()
{

std::cout << "\n";
std::cout << "Employee ID Number: " << Employee::getIdNum() << "\n";
std::cout << "Name: " << Employee::getLastName() <<", " << Employee::getFirstName() << " " << Employee::getMiddleInitial() <<"." << "\n";
std::cout << "Department Code: " << Employee::getDeptCode() << "\n";
std::cout << "Total Hours Worked : " << HourlyEmployee::getTotalHoursWorked() << "\n";
std::cout << "Hourly Rate : " << HourlyEmployee::getHourlyRate() << "\n";
std::cout << "Employee Weekly Payment : " << HourlyEmployee::calculatedHourlyPayment() << "\n";
}
```

Figure 9. hourlyEmployee.hpp Class File

hourlyEmployee.hpp Header File

```
/* File: hourlyEmployee.hpp */

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

class HourlyEmployee: public Employee{

public:

        HourlyEmployee(long = 0, const string & ="" , const string & ="", const string & = "", int =0, int=0, int=0);
        int totalHours;
        int hourlyRate;
        int overtime;
        double WeeklyPayment;
        void setTotalHoursWorked(int totalHours);
        int getTotalHoursWorked();
        void setHourlyRate(int hourRate);
        int getHourlyRate();
        double calculatedHourlyPayment();
        void printFunctionforHourlyEmpl();

};
```

Figure 10. hourlyEmployee.hpp Header File

Test.cpp - Test File

```
/* File: test.cpp */
// File to test the basic employee class

#include<iostream>
#include <string>
#include "employee.hpp"
#include "salariedEmployee.hpp"
#include "hourlyEmployee.hpp"


using namespace std;

int main()
{

std::cout << "\n";
std::cout << "Print Employee 1,2,3 Salary Information: ";
std::cout << "\n";
 // full time = 28000, half-time: 14000
 SalariedEmployee employee1(001, "Jones", "Booker", "T", 22, 14000); // half time employee
 SalariedEmployee employee2 (002, "Hendrix", "Jimi", "NMI ", 14, 32000);
 SalariedEmployee employee3 (003, "Morrison", "Jim", "D", 03, 28000); // full time employee

 employee1.printFunctionforSalariedEmpl();
 employee2.printFunctionforSalariedEmpl();
 employee3.printFunctionforSalariedEmpl();

std::cout << "\n";
std::cout << "Print Employee 4,5,6 Hourly Information: ";
std::cout << "\n";
//(long id, const std::string &lastName, const std::string &firstName, const std::string &initial,int deptCode, int totalHrs, double hourRate);
// full time = 40 hrs, half time = 20 hours
 HourlyEmployee employee4(004, "Richardson", "Samantha", "M", 11, 34, 12);
 HourlyEmployee employee5(005, "Smith", "Laura", "K", 67, 40, 14); // full time employee
 HourlyEmployee employee6(006, "Lavisky", "Alexandra", "S", 05, 20, 14); // half time employee

 employee4.printFunctionforHourlyEmpl();
 employee5.printFunctionforHourlyEmpl();
 employee6.printFunctionforHourlyEmpl();
```

Figure 11. Test.cpp Test file for Employee Base Class

```
 employee4.printFunctionforHourlyEmpl();
 employee5.printFunctionforHourlyEmpl();
 employee6.printFunctionforHourlyEmpl();

std::cout << "\n";
std::cout << "Print Employee 1 and 5 information using printEmployee() from Employee.cpp: ";
std::cout << "\n";
employee1.printEmployee();
employee5.printEmployee();

return 0;
}
```

Figure 12. Test.cpp Test file for Employee Base Class

Test.cpp File Output Results to Test Employee Base Class

```
(base) clarizza@MacBook-Pro ECE-330-SoftwareDesign-HW4 % ./run_employee

Print Employee 1,2,3 Salary Information:

Employee ID Number: 1
Name: Jones , Booker, T
Department Code: 22
Employee Salary: 14000

Employee ID Number: 2
Name: Hendrix , Jimi, NMI
Department Code: 14
Employee Salary: 32000

Employee ID Number: 3
Name: Morrison , Jim, D
Department Code: 3
Employee Salary: 28000

Print Employee 4,5,6 Hourly Information:

Employee ID Number: 4
Name: Richardson, Samantha M.
Department Code: 11
Total Hours Worked : 34
Hourly Rate : 12
Employee Weekly Payment : 408

Employee ID Number: 5
Name: Smith, Laura K.
Department Code: 67
Total Hours Worked : 40
Hourly Rate : 14
Employee Weekly Payment : 560

Employee ID Number: 6
Name: Lavisky, Alexandra S.
Department Code: 5
Total Hours Worked : 20
Hourly Rate : 14
Employee Weekly Payment : 280
```

Figure 13. Test.cpp output results from Employee base class, salaried and hourly employee Classes

```
Employee ID Number: 6
Name: Lavisky, Alexandra S.
Department Code: 5
Total Hours Worked : 20
Hourly Rate : 14
Employee Weekly Payment : 280

Print Employee 1 and 5 information using printEmployee() from Employee.cpp:

Employee ID Number: 1
Name: Jones, Booker T.
Dept Code: 22

Employee ID Number: 5
Name: Smith, Laura K.
Dept Code: 67
```

Figure 14. Test.cpp output resuslts from Employee base class, salaried and hourly employee Classes

Conclusion

Overall, working through each of the classes salariedEmployee() and hourlyEmployee() was very interesting and fun. I had heard of inheritance before and used it only once, but with this homework assignment even though I had some blocks in the way I was able to learn and understand how inheritance works and how it is possible to implement functions or methods defined in a different class in a new class. In this way we are able to make our program more efficient, save memory space, and save time. I also learned to use header files in the right way as well as learned how the conventions for each type of file work. I had some trouble at the begging since I was importing the .cpp files into some of my other .cpp or my .hpp files, including the test.cpp. However, after looking at the lecture again and research on the c++ documentation I was able to get the file convention correctly and therefore import the correct and only necessary files.