**C++ DERIVED CLASS LAB REPORT**

**1) Enter your name, student ID, platform (Mac or PC) and date**

Name and Student ID: Samuel Indurkar, 0888068

Class: CIS054 C/C++ Programming

Platform (Mac or PC): Eclipse and g++ on MAC  
Date: 7/18/2017

**DESCRIPTION:**Using the Derived-Class.cpp file, create an additional derived class (sub class) of type **Employee**. The Employee class must be derived from the **Person** class and include the following data items:  
 int EmpNo; // Employee number

double Hours;

double PayRate;

The print method for **Employee** needs to compute the pay based on Hours and PayRate and then display the **Name,** and **Pay.**

Update the **main** program to include at least two employees.

**LAB REPORT:  
2) Determine the Inputs, Processing and Outputs before creating the program**

|  |  |  |
| --- | --- | --- |
| **INPUTS** | **PROCESSING** | **OUTPUTS** |
| Pre-defined input of Student, Teacher and Employee. | Person is the base class with Print() member routine. All other derived classes are derived from it. Each derived class prints its own specific print. In case of Employee, it calculates the pay as hours \* pay\_rate | Each derived class prints its own specific print for Student, Teacher and Employee |

**3) Fill in the EXPECTED & ACTUAL RESULTS**

|  |  |  |
| --- | --- | --- |
| **TEST DATA VALUES**  Provide several different examples of menu selections | **EXPECTED RESULT**  Computed values before the program is run | **ACTUAL RESULT**  Fill in the output displayed  by the program |
| Student s1("Joe Williams", 12);  Student s2("Mary Smith ", 9);  Student s3("Tam Nguyen ", 10);  Student s4("Jose Chavez ", 11);  Teacher t1("Dan McElroy ", 28, 3);  Teacher t3("Fred Jones ", 18, 2);  Employee e1("Joe Indurkar", 1234, 8, 15);  Employee e2("Jane Johnson", 5678, 8, 12); | Teacher: Dan McElroy Students: 28 Classes: 3  Student: Mary Smith Units: 9  Student: Tam Nguyen Units: 10  Student: Jose Chavez Units: 11  Student: Mary Smith Units: 9  Teacher: Fred Jones Students: 18 Classes: 2  Employee: Joe Indurkar EmpNo: 1234 Pay: 120  Employee: Jane Johnson EmpNo: 5678 Pay: 96 | Teacher: Dan McElroy Students: 28 Classes: 3  Student: Mary Smith Units: 9  Student: Tam Nguyen Units: 10  Student: Jose Chavez Units: 11  Student: Mary Smith Units: 9  Teacher: Fred Jones Students: 18 Classes: 2  Employee: Joe Indurkar EmpNo: 1234 Pay: 120  Employee: Jane Johnson EmpNo: 5678 Pay: 96 |

**DISCUSSION:**

**4) Complete the DISCUSSION section. It does not need to be long, but it needs to be complete.**4a) What did you do to develop the program? ("Followed the Directions" is not a complete description)

The program takes a pre-defined input of derived class called Student, Teacher and Employee which is derived from base class Person. Person is the base class with Print() member routine. All other derived classes are derived from it. Each derived class prints its own specific print. In case of Employee, it calculates the pay as hours \* pay\_rate

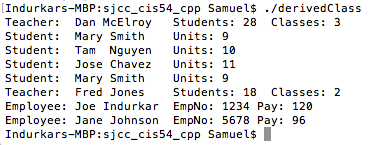
4b) What problems did you have and how did you overcome the problems?

No problem.

**PROGRAM OUTPUT:**

**5) Show screen shots for at least TWO sample purchases of the program execution.**

Refer to previous lab assignments for instructions on how to capture a screen or portions of a screen for either the PC or a Mac

****

**PROGRAM LISTING:**

**6) Copy and paste the code that YOU typed to make the program work. Your program should include a comment block at the top that shows the name of the program, date, version and your name.**

/\*

\* derivedClass.cpp

\*

\* Created on: Jul 18, 2017

\* Author: Samuel

\*/

// Derived-Class.cpp : Defines the entry point for the console application.

//

#include <iostream>

#include <cstring>

using namespace std;

////////////// Person class Definition /////////////////

class Person {

protected: // A derived class can access protected data

char Name[20];

public:

Person (char\* n)

{

strcpy(Name, n);

}

virtual void print() const

{

cout << "Person: " << Name << endl;

}

};

///////////// Student class Definition //////////////

class Student : public Person {

private:

int units;

public:

// Student constructor gets name from Person class

Student (char\* n, int u) : Person ( n)

{

units = u;

}

virtual void print() const

{

cout << "Student: " << Name << " Units: " << units << endl;

}

};

//////////// Teacher class Definition /////////////////

class Teacher : public Person {

private:

int numberOfStudents;

int numberOfClasses;

public:

// Teacher constructor gets 'name' from the Person class

Teacher (char\* n, int s, int c) : Person ( n)

{

numberOfStudents = s;

numberOfClasses = c;

}

virtual void print() const

{

cout << "Teacher: " << Name

<< " Students: " << numberOfStudents

<< " Classes: " << numberOfClasses << endl;

}

};

///////////// Employee class Definition //////////////

class Employee : public Person {

private:

int EmpNo; // Employee number

double Hours;

double PayRate;

public:

// Employee constructor gets name from Person class

Employee (char\* n, int id, int hr, int pRate) : Person ( n)

{

EmpNo = id;

Hours = hr;

PayRate = pRate;

}

virtual void print() const

{

double pay = Hours \* PayRate;

cout << "Employee: " << Name << " EmpNo: " << EmpNo << " Pay: " << pay << endl;

}

};

/////////// main program ///////////////////////

int main(int argc, char\* argv[])

{

// create objects from several different types of classes

Student s1("Joe Williams", 12);

Student s2("Mary Smith ", 9);

Student s3("Tam Nguyen ", 10);

Student s4("Jose Chavez ", 11);

Teacher t1("Dan McElroy ", 28, 3);

Teacher t3("Fred Jones ", 18, 2);

Employee e1("Joe Indurkar", 1234, 8, 15);

Employee e2("Jane Johnson", 5678, 8, 12);

// Create an array of pointers to different people

Person \*List[] = { &t1, &s2, &s3, &s4, &s2, &t3, &e1, &e2 };

int SizeOfList = sizeof(List)/sizeof(Person\*);

for (int i=0; i<SizeOfList; i++)

{

// call the appropriate virtual print routine based

// on the type of object being pointed to

List[i]->print();

}

return 0;

}