Object Oriented Programming Lab

Lab 11 **Marks 10**

Instructions

Work on this lab individually. You can use your books, notes, handouts etc. but you are not allowed to borrow anything from your peer student. *You are strictly **NOT ALLOWED** to include any additional data-members/functions/constructors* in your class.

Marking Criteria

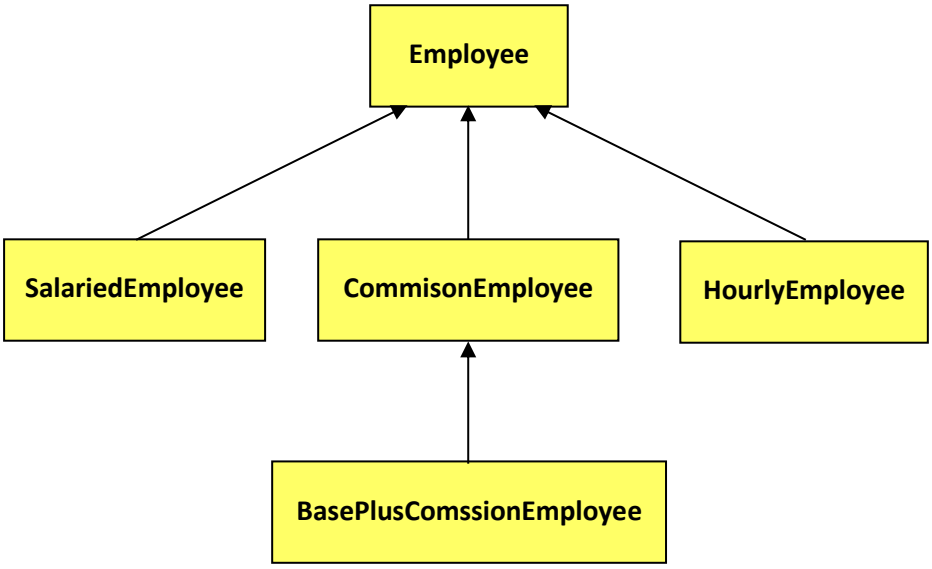
Show your work to the instructor before leaving the lab to get some or full credit.

What you must do

Program the following task in your C++ compiler and then compile and execute them. *Write the **main** function first and keep testing the functionality of each function once created.*

Employee Inheritance Hierarchy

Implement the following class hierarchy, the inheritance access level should be **public** for all classes.



Employee Class Details

- Declare three data members named **firstname**, **lastname** and **SSN** of type **string** with **private access**.
- Implement a **parameterized constructor**.
- Implement **get/set** function for all data members.
- Implement a **virtual function** named **print** that displays the **name** and **social security number** of a particular employee.
- Implement a **pure virtual function** named **earnings** that **calculates and return the earning** of a particular employee.

SalariedEmployee Class Details

- Declare a data member named **weeklySalary** of type **double** with **private access**.
- Implement a **parameterized constructor**, which initializes all the data members of **SalariedEmployee** with **default parameter** set to **0 for weeklySalary**.
- Implement **get/set** function for all data members.
- Implement a **virtual function** named **print** that displays the **name**, **social security number** and **weekly salary** of a particular employee.
- Implement a **virtual function** named **earnings** that **return the earning** of a particular salaried employee.

HourlyEmployee Class Details

- Declare two data members named **wage** and **hours** of type **double** with **private access**.
- Implement a **parameterized constructor**, which initializes all the data members of **HourlyEmployee** with **default parameter** set to **0** for **wage** and **hours**.
- Implement **get/set** function for all data members.
- Implement a **virtual function** named **print** that displays the **name**, **social security number**, **wage** and **hours** of a particular employee.
- Implement a **virtual function** named **earnings** that **calculates and return** the **earning** of a particular hourly employee. The salary can by calculated by multiplying hours with wage.

CommissionEmployee Class Details

- Declare two data members named **grossSales** and **commissionRate** of type **double** with **private access**.
- Implement a **parameterized constructor**, which initializes all the data members of **CommissionEmployee** with **default parameter** set to **0** for **grossSales** and **commissionRate**.
- Implement **get/set** function for all data members.
- Implement a **virtual function** named **print** that displays the **name**, **social security number**, **gross sales** and **commission rate** of a particular employee.
- Implement a **virtual function** named **earnings** that **calculates and return** the **earning** of a particular commissioned employee. The salary can by calculated by multiplying commission rate with gross sales.

BasePlusCommissionEmployee Class Details

- Declare data members named **baseSalary** of type **double** with **private access**.
- Implement a **parameterized constructor**, which initializes all the data members of **BasePlusCommissionEmployee** with **default parameter** set to **0** for **baseSalary**.
- Implement **get/set** function for all data members.
- Implement a **virtual function** named **print** that display the **name**, **social security number**, **gross sales**, **commission rate** and **base salary** of a particular employee.
- Implement a **virtual function** named **earnings** that **calculates and return the earning** of a particular base plus commissioned employee. The salary can be calculated by adding the **CommissionEmploy::earnings + baseSalary**.

Main Program Details

1. Create objects of each class created above with relevant information and display the personal information of each employee with their salaries.

2. Now create a **pointer array of type *Employee* with size of 4**, each location of this array should point to object of **SalariedEmployee**, **HourlyEmployee**, **CommissionEmployee** and **BasePlusCommsissionEmployee** created above.

3. Now loop through the entire pointer array and display the information of each employee using only two statements

```
eptr[i] -> print();
eptr[i] -> earnings();
```

   where **eptr** is a pointer array of type **Employee**.

☺ ☺ ☺ **BEST OF LUCK** ☺ ☺ ☺
