Data Structure and Advanced Programming

Homework #1

Due: 2021/3/10 11:59pm (CST)

NOTE: Please upload your C++ source codes (by copy-paste) to PDOGS before the due date and time.

The goal of this homework is to practice what you learn about inheritance and polymorphism. In this homework, you will implement a C++ program with four classes: Staff, Manager, Engineer, and Sales. Staff is a base (parent) class. Manager, Engineer, and Sales are derived (child) classes that inherited Staff. The class Staff contains the following members:

1. void set_min_salary()
2. void set_seniority()
3. int get_min_salary()
4. virtual int total_salary() = 0
5. int _min_sal, _seniority

On the other hand, the class Manager, Engineer, and Sales only contain one member :

1. int total_salary()

Moreover, the way of computing the total salary is different for the class Manager, Engineer, and Sales, which means that the implementations of total_salary() are different in these classes.

- Manager's total salary : min_sal + min_sal * seniority
- Engineer's total salary : (min_sal + min_sal * seniority) / 2
- Sales' total salary : min_sal + (min_sal * seniority) / 2

There will be four inputs to your program.

1. Staff's minimum salary (int)
2. Manager's seniority (int)
3. Engineer's seniority (int)
4. Sales' seniority (int)

For the input, the program should be able to read multiple lines and one line containing these four integers followed by the above order and separated by one space. Please see the program structure and Sample Input / Output for more information.

Program Structure

Please use the following program in your .cpp file and complete the parts labeled by `// YOUR CODE HERE` to make your program fulfill the requirements mentioned above and can successfully output the correct results.

```
class Staff {
  // YOUR CODE HERE
  };
class Manager : public Staff {
  // YOUR CODE HERE
  };
class Engineer : public Staff {
  // YOUR CODE HERE
  };
class Sales : public Staff {
```

```
  // YOUR CODE HERE
  };
int main() {
  int minsalary, M_seniority, E_seniority, S_seniority;
  Manager M1;
  Engineer E1;
  Sales S1;
  Staff * st1 = &M1;
  Staff * st2 = &E1;
  Staff * st3 = &S1;

  while(cin>>minsalary){
      cin>>M_seniority;
      cin>>E_seniority;
      cin>>S_seniority;
      // YOUR CODE HERE
      cout << st1->get_min_salary() << endl;
      cout << st2->get_min_salary() << endl;
      cout << st3->get_min_salary() << endl;
      cout << st1->total_salary() << endl;
      cout << st2->total_salary() << endl;
      cout << st3->total_salary() << endl;

  }
  return 0;
}
```

Sample Input / Output

Take the input (1000 10 3 5) for example, the first integer 10000 stands for Staff's minimum salary, 10 stands for Manager's seniority, 3 stands for Engineer's seniority, and 5 is Sales' seniority.

// Input
10000 10 3 5
 	
// Output
10000
10000
10000
110000
20000
35000
	// Input
	22000 7 11 15
	51000 8 3 11
	16000 3 4 15
	
	// Output
	22000
	22000
	22000
	176000
	132000
	187000
	51000
	51000
	51000
	459000
	102000
	331500
	16000
	16000
	16000
	64000
	40000
	136000