You are given the following class definition

```
    class Person
    {
    public:
    Person(string); // initialise the name
    virtual void printname();
    protected:
    string name;
    };
```

1. Add to this another 2 classes, `Employee` and `Customer`, each of which are derived from `Person`.

2. Each of these classes also has a `printname()` method, which are implemented differently:

– `printname` method of `Employee` class will print out the name and salary (which is an extra data member of the `Employee` class).

– `printname` method of `Customer` class will print out the name and a message saying they want to make a complaint.

3. Define constructors for each of these classes which pass back a name variable to the `Person` class for initialisation. The constructor of the `Employee` class should, in addition, initialise the salary, which is an attribute of that class.

4. In your main program, declare a pointer to `Person` class – call it `personPtr`. This is the base class pointer.

– Create and initialise a `Person` object and call its `printname` method via the pointer.

– Create and initialise an `Employee` object, and call its `printname` method via the base class pointer.

– Create and initialise a `Customer` object, and call its `printname` method via the base class pointer.

```
Sample Output
My name is Mark
My name is Tom and my salary is 34, 000
My name is Ed and I want to make a complaint
```

5. Now take out the virtual keyword in the `Person` class and re-run the program to see how the output differs.

6. Rewrite the `Person` class so that it is an abstract class, and see what difference this makes to your program.

7. Now add appropriate overloaded operators as member functions to the `Person` class which will enable you to compare and order people according to their name. (*Look at the compare method of the string class http://www.cplusplus.com/reference/string/string/*)

8. Add code to your main method to enable you to create 3 (or more) `Person` objects, store pointers to them in an array of pointers to `Person`, and order them alphabetically.