

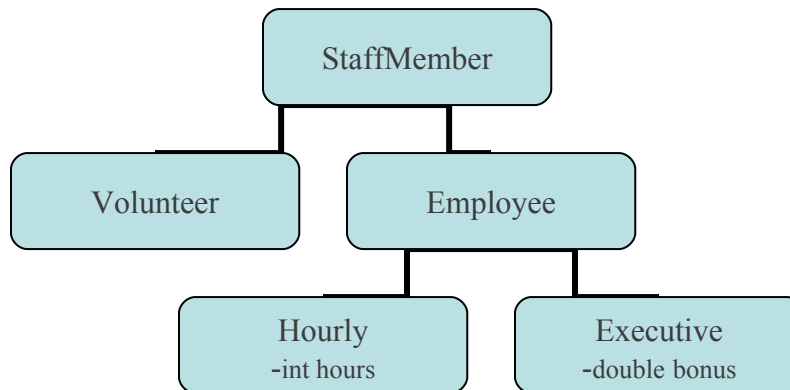
Write a program that starts with an Abstract class to represent a general StaffMember of a corporation. Abstract classes are not instantiated and the StaffMember class does not represent any particular type of employee. It represents the ancestor of all employee classes that are to be created, and it contains base information that applies to all types of employees.

A StaffMember class must have the following members:

Data: name, address, phone (all are Strings and protected).

Methods: 1. A constructor with all three strings as input.
2. toString that returns a String with all three data types.
3. An abstract method called pay, that is public, and returns the pay. The pay is a double value.

Now create two classes that directly inherit the StaffMember class. Call these classes Volunteer, and Employee. Two more classes, Hourly and Executive directly extend the Employee class. The figure below shows the inheritance scheme for this project.



Class Volunteer has its own constructor and requires the same data the StaffMember constructor requires, and a volunteer gets a pay of 0.0. The class Employee has its own constructor that requires two additional data items; social security number (String), and a payRate (double). Employee has a method setPay(double) that can set a new value into its payRate field.

Class Executive extends Employee and it has an additional private data member called bonus (double). Executive has a mutator method called awardBonus that sets the bonus value for the class. It also has a method named pay that returns the pay by calling its super class pay method and adding a bonus. It then resets bonus = 0 before returning.

Class Hourly extends Employee. It has a private data member called hoursWorked (double), and it requires the same parameters that the Employee constructor requires. The constructor also sets hoursWorked to 0.0. The Hourly class has a mutator method that takes a double value as input and adds this value to the instance variable hoursWorked. This class also has a method to return the pay (hours worked times the payrate), and its own toString method that will print out all

information about the employee. It contains a method setPay(double) that sets a new value for the payrate.

Your application should start by call the constructor for a class called Staff that creates 10 employees and puts them in an array of StaffMember. See below for the initial set up.

The Staff class should then do:

Create a staff of ten as shown below, and provide a method that goes through the array and print all relevant information on the employee and their pay.

```
public class Staff
{
    private StaffMember[] staffList;
    //-----
    // Sets up the list of staff members.
    //-----
    public Staff ()
    {
        staffList = new StaffMember[10];
        staffList[0] = new Executive ("Sam", "123 Main Line",
            "555-0469", "123-45-6789", 2423.07);
        staffList[1] = new Employee ("Carla", "456 Off Line",
            "555-0101", "987-65-4321", 1246.15);
        staffList[2] = new Employee ("Woody", "789 Off Rocker",
            "555-0000", "010-20-3040", 1169.23);
        staffList[3] = new Hourly ("Diane", "678 Fifth Ave.",
            "555-0690", "958-47-3625", 10.55);
        staffList[4] = new Volunteer ("Norm", "987 Suds Blvd.",
            "555-8374");
        staffList[5] = new Volunteer ("Cliff", "321 Duds Lane",
            "555-7282");
        staffList[6] = new Employee ("Coach", "1 Coach Court",
            "555-2345", "010-20-3458", 1100.23);
        staffList[7] = new Hourly ("Daffy", "678 Mallard Court.",
            "555-7690", "128-47-3625", 6.55);
        staffList[8] = new Hourly ("Bugs", "10 Hole In Ground",
            "555-1111", "111-22-3333", 5.00);
        staffList[9] = new Hourly ("Elmer", "22 Fudd Way",
            "555-7282", "222-33-1111", 6.25);
    }

    //-----
    // method payday pays all staff members.
    //-----
    public void payday ()
    {
        double amount;

        for (int count=0; count < staffList.length; count++)
        {
            System.out.println (staffList[count]);
        }
    }
}
```

```

        amount = staffList[count].pay(); // polymorphic

        if (amount == 0.0)
            System.out.println ("Thanks!");
        else
            System.out.println ("Paid: " + amount);

        System.out.println ("-----");
    }
}
}

```

The class containing your main method can mirror the class Firm shown below. So in all your project needs to create the class Firm and the class Staff.

```

//*****
// Main program application:
// name: Firm
// purpose: Creates a Staff for an organization and generates the payroll report.
//*****

```

```

public class Firm
{
    //-----
    // Creates a staff of employees for a firm and pays them.
    //-----
    public static void main (String[] args)
    {
        Staff personnel = new Staff();

        // awardBonus of 500.00 to Sam;
        // add 40 hours to Diane, add 36 hours to Daffy, and 30 hours to Bugs,
        // add 28 hours for Elmer.
        // give bugs a raise to 5.50, give Coach a raise to 1550.00

        personnel.payday();
    }
}

```

Document all of your classes and your application by inserting comments in your code, including pre-conditions and post-conditions for each method, your name section number, and assignment number. Turn in your program according to the usual submission guidelines.