1. Fill in some code for an Employee class, following guidelines given through comments.

```java
public class Employee {
    // instance variables
    String firstName;
    String lastName;
    int yearHired

    // constructor: Construct object with x, y, and z as first
    //              name, last name, and year hired, respectively.
    public Employee (String firstName, String lastName, int yearHired)
    {
        this.firstName = firstName;
        this.lastName = lastName;
        this.yearHired = yearHired;
    }

    // toString(): Returns a String corresponding to object.
    public String toString()
    {
        return "Employee name: \" + firstName + \" \" + lastName + \" hired in \" + yearHired + \" year.\"
    }

    // getYearHired(): Returns the year this Employee was hired.
    public int getYearHired()
    {
        return yearHired;
    }
}
```

2. Write client code that uses the Employee class:
   a) Instantiate an Employee object for someone named Alexa Gomez, hired in 2014. Assign it to a variable named coderBoss.

   ```java
coderBoss = new Employee("Alexa", "Gomez", 2014);
```

   b) Suppose you have two Employee objects e1, e2. Write some client code that uses the getYearHired() and toString() methods of the Employee class to print the information associated with the Employee who has been with the company the longest (i.e., was hired earlier).

   ```java
   if (e1.getYearHired() < e2.getYearHired())
   {
       System.out.println(e1.toString());
   }
   else
   {
       System.out.println(e2.toString());
   }
   ```
1. Fill in some code for an Employee class, following guidelines given through comments.

```java
public class Employee {
    // instance variables
    String name;
    String position;
    double hourly; // hourly wages
    double hours;  // hours worked this week

    // constructor
    public Employee(String x, String y, double w, double h) {
        name = x;
        position = y;
        hourly = w;
        hours = h;
    }

    // wages(): Returns the wages of employee for this week
    // calculated as 1.5 of hourly rate for hours over 40.
}
```

2. Write client code that uses the Employee class:
   a) Instantiate an Employee object with name “Lucia Taylor” position “software engineer”, with hourly rate $42.50 and who worked 52.5 hours last week. Assign it to a variable named coder.

   ```java
   Employee coder = new Employee("Lucia Taylor", "software engineer", 42.50, 52.5);
   ```

   b) Suppose you have three Employee objects e1, e2 and e3. Write some client code that uses the Employee class to calculate and print the average of their wages.
   (Note: it is NOT necessary to format as currency).