```
/* This program demonstrates use of inheritance and polymorphism.  A class Employee is
 * defined, with two subclass representing hourly and salaried employees.
 *
 * Copyright (c) 2000, 2001, 2004 - Russell C. Bjork
 */

import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import java.text.DecimalFormat;

/** Abstract base class
 */
abstract class Employee
{
    /** Constructor
     *   @param name the employee's name
     *   @param ssn the employee's Social Security number
     */
    protected Employee(String name, String ssn)
    {
        this.name = name;
        this.ssn = ssn;
    }
    /** Accessor for name
     *   @return the employee's name
     */
    public final String getName()
    {
        return name;
    }
    /** Accessor for ssn
     *   @return the employee's social security number
     */
    public final String getSSN()
    {
        return ssn;
    }
    /** Generate String containing all information on employee
     *   @return a string containing all the information on this employee
     */
    public String toString()
    {
        return name + " SSN: " + ssn;
    }
    /** Calculate weekly pay - implemented by each subclass
     *   @return the amount of the check to be printed for this employee
     */
    public abstract double weeklyPay();

    // Instance variables - the employee's name and Social Security number

    private String name, ssn;
}
```

```
/** Representation for hourly employees
 */
class HourlyEmployee extends Employee
{
    /** Constructor
     *
     *   @param name the employee's name
     *   @param ssn the employee's Social Security number
     *   @param hourlyRate the employee's hourly pay rate
     */
    public HourlyEmployee(String name, String ssn, double hourlyRate)
    {
        super(name, ssn);
        this.hourlyRate = hourlyRate;
    }

    /** Generate String containing all information on employee
     *
     *   @return a string containing all the information on this employee
     */
    public String toString()
    {
        String commonInfo = super.toString();
        return commonInfo + " Hourly Rate: $" +
            (new DecimalFormat("0.00")).format(hourlyRate);
    }

    /** Calculate weekly pay.  The user is asked to input the number of
     *   hours the employee worked
     *
     *   @return the amount of the check to be printed for this employee
     */
    public double weeklyPay()
    {
        String hoursWorkedString = JOptionPane.showInputDialog(
            null, "Hours worked for " + getName() + "?");
        double hoursWorked = Double.parseDouble(hoursWorkedString);

        if (hoursWorked > 40)
            return hourlyRate * 40 + hourlyRate * 1.5 * (hoursWorked - 40);
        else
            return hourlyRate * hoursWorked;
    }

    // Instance variable - the employee's hourly pay rate

    private double hourlyRate;
}
```

```
/** Representation for salaried employees
 */
class SalariedEmployee extends Employee
{
    /** Constructor
     *
     *  @param name the employee's name
     *  @param ssn the employee's Social Security number
     *  @param annualSalary the employee's annual salary
     */
    public SalariedEmployee(String name, String ssn, double annualSalary)
    {
        super(name, ssn);
        this.annualSalary = annualSalary;
    }

    /** Generate String containing all information on employee
     *
     *  @return a string containing all the information on this employee
     */
    public String toString()
    {
        String commonInfo = super.toString();
        return commonInfo + " Salary: $" +
            (new DecimalFormat("0.00")).format(annualSalary);
    }

    /** Calculate weekly pay.  The hours worked this week is irrelevant for
     *  a salaried employee, who is always paid the same amount.
     *
     *  @return the amount of the check to be printed for this employee
     */
    public double weeklyPay()
    {
        return annualSalary / 52;
    }

    // Instance variable

    private double annualSalary;
}
```

```
/** Tester for the above
 */
class EmployeeTester
{
    public static void main(String [] args)
    {
        // Ask user for number of employees; create array of appropriate size
        int numberOfEmployees = Integer.parseInt(
            JOptionPane.showInputDialog(null, "Employee Count? "));
        Employee [] employees = new Employee [numberOfEmployees];
        // Read information on individual employees.  In each case, first ask
        // user what kind of employee, then pop up dialog to read relevant
        // information
        String [] employeeKinds = { "Hourly", "Salaried" };
        String [] hourlyLabels = { "Name", "SSN", "Hourly Rate" };
        String [] salariedLabels = { "Name", "SSN", "Annual Salary" };
        for (int i = 0; i < numberOfEmployees; i ++)
        {
            String kind = (String) JOptionPane.showInputDialog(null,
              "What kind of employee is employee " + (i+1) + "? ",
              "", JOptionPane.QUESTION_MESSAGE,null, employeeKinds, "Hourly");
            if (kind.equals("Hourly"))
            {
                 String [] employeeInfo = MultiInputPane.showMultiInputDialog(
                     null, hourlyLabels, "Employee Information");
                 employees[i] = new HourlyEmployee(employeeInfo[0],
                     employeeInfo[1], Double.parseDouble(employeeInfo[2]));
            }
            else
            {
                 String [] employeeInfo = MultiInputPane.showMultiInputDialog(
                     null, salariedLabels, "Employee Information");
                 employees[i] = new SalariedEmployee(employeeInfo[0],
                     employeeInfo[1], Double.parseDouble(employeeInfo[2]));
            }
        }
        // Print out information on all the employees
        System.out.println("Information on employees:"); System.out.println();
        for (int i = 0; i < employees.length; i ++)
            System.out.println(employees[i].toString());

        // Pay them.  Note that, for each hourly employee, the user will be
        // asked to input the hours worked

        System.out.println();
        System.out.println("Weekly payroll:"); System.out.println();
        for (int i = 0; i < employees.length; i ++)
            System.out.println(employees[i].getName() + " $" +
                (new DecimalFormat("0.00")).format(employees[i].weeklyPay()));

        System.exit(0);
    }
}
```