

## Answer on Question #45264, Programming, Java | JSP | JSF

### Problem.

Write a program to print the pay slip for each employee. Your program must use the properties of object-oriented programming which are polymorphism and inheritance in constructing the classes for the employees. Your program must have a menu that gives the user an option whether to process the pay slip for part time or full time employee. For a part time employee, the input will be hours worked, hourly rate and sales achieved, and for full time employee, it will be overtime rate, overtime hours worked and sales achieved. Assume the basic pay for full time is \$2000.

Below is the commission rate for the sales achieved.

\$250 - \$600 = 2%

\$601 - \$900 = 4%

Over \$900 = 6%

### Solution.

*Code (Main.java)*

```
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        float time;
        float rate;
        float sales;
        FullTimeEmployee employee = null;

        System.out.println("(a) Full time employee.");
        System.out.println("(b) Part time employee.");
        Scanner myScanner = new Scanner(System.in);
        char choice = myScanner.next().charAt(0);

        // Input
        if (choice == 'a') {
            System.out.print("Time: ");
            time = myScanner.nextFloat();

            System.out.print("Rate: ");
            rate = myScanner.nextFloat();

            System.out.print("Sales: ");
            sales = myScanner.nextFloat();

            employee = new FullTimeEmployee(time, rate, sales);
        } else if (choice == 'b') {
            System.out.print("Time: ");
            time = myScanner.nextFloat();

            System.out.print("Rate: ");
            rate = myScanner.nextFloat();

            System.out.print("Sales: ");
            sales = myScanner.nextFloat();

            employee = new PartTimeEmployee(time, rate, sales);
        } else {
            System.out.println("Incorrect input!");
        }

        // Output
```

```

        System.out.println(employee.print());
    }
}

```

### Code (FullTimeEmployee.java)

```

public class FullTimeEmployee {
    protected float time;
    protected float rate;
    protected float sales;
    protected float income;
    private static float BASIC_PAY = 2000;

    /**
     * Default constructor
     */
    public FullTimeEmployee() {

    }

    /**
     * Custom constructor, set parameters.
     *
     * @param time overtime hours worked
     * @param rate overtime rate
     * @param sales sales achieved
     */
    public FullTimeEmployee(float time, float rate, float sales) {
        this.time = time;
        this.rate = rate;
        this.sales = sales;
        income = BASIC_PAY;
        income += time * rate;
        if ((250 <= sales) && (sales <= 600)) {
            income += sales * 0.02;
        }
        if ((601 <= sales) && (sales <= 900)) {
            income += sales * 0.04;
        }
        if (sales > 900) {
            income += sales * 0.06;
        }
    }

    /**
     * Print pay slip.
     *
     * @return pay slip, as string.
     */
    public String print() {
        return "To pay: $" + Float.toString(income);
    }
}

```

### Code (PartTimeEmployee.java)

```

public class PartTimeEmployee extends FullTimeEmployee {
    /**
     * Custom constructor, set parameters.
     *
     * @param time hours worked
     * @param rate hourly rate
     * @param sales sales achieved
     */
    public PartTimeEmployee(float time, float rate, float sales) {
        this.time = time;
        this.rate = rate;
    }
}

```

```

        this.sales = sales;
        income += time * rate;
        if ((250 <= sales) && (sales <= 600)) {
            income += sales * 0.02;
        }
        if ((601 <= sales) && (sales <= 900)) {
            income += sales * 0.04;
        }
        if (sales > 900) {
            income += sales * 0.06;
        }
    }
}

```

### **Result**

- (a) Full time employee.
- (b) Part time employee.

a

Time: 100

Rate: 10

Sales: 300

To pay: \$3006.0