

Ex.No: 3	Employee Payroll System
Date:	

Aim:

To create a Java console application for employee payroll process management. This project includes Employee and they further classified as Programmer, Assistant Professor, Associate Professor, Professor.

Algorithm:

- Step 1 Start the process
- Step 2 Prompt the user with choice 1. Programmer 2. Assistant Professor 3. Associate Professor 4. Professor 5. Exit and get the choice.
- Step 3 If user selects a Programmer then proceed to step 4
- Step 4 Get the user details [Name, ID, Address, Mail ID and Mobile Number] and goto step 5
- Step 5 Proceed to Programmers Payment Processing
 - Step 5.1 Get the basic pay of Programmer
 - Step 5.2 If user enters -1 assume basic pay as 30000 and goto step 15
 - Step 5.3 Else directly go to step 15
- Step 6 If user selects Assistant Professor step 7
- Step 7 Get the user details [Name, ID, Address, Mail ID and Mobile Number] and goto step 8
- Step 8 Proceed to Assistant Professor Payment Processing
 - Step 8.1 Get the basic pay of Assistant Professor
 - Step 8.2 If user enters -1 assume basic pay as 25000 and goto step 15
 - Step 8.3 Else directly go to step 15
- Step 9 If user selects Associate Professor step 10
- Step 10 Get the user details [Name, ID, Address, Mail ID and Mobile Number] and goto step 11
- Step 11 Proceed to Associate Professor Payment Processing
 - Step 11.1 Get the basic pay of Associate Professor
 - Step 11.2 If user enters -1 assume basic pay as 40000 and goto step 15
 - Step 11.3 Else directly go to step 15
- Step 12 If user selects Professor step 13
- Step 13 Get the user details [Name, ID, Address, Mail ID and Mobile Number] and goto step 14
- Step 14 Proceed to Professor Payment Processing
 - Step 14.1 Get the basic pay of Professor
 - Step 14.2 If user enters -1 assume basic pay as 70000 and goto step 15
 - Step 14.3 Else directly go to step 15
- Step 15 Compute $\text{Per_Day_Pay} = \text{original_basic_pay} / \text{no_of_days_in_the_current_month}$
- Step 16 Get the number of days worked from user that include Cl, WH, FH and exclude the LOP

- Step 17 Check `no_days_worked <= no_of_days_in_the_current_month`. Else display “Error Message” and goto step 18
- Step 18 Compute `Current_Basic_Pay = Per_Day_Pay * no_days_worked`
- Step 19 Compute Following and Store
- DA = $(\text{Current_Basic_Pay}/100) * 97$
- HRA = $(\text{Current_Basic_Pay}/100) * 12$
- PF = $(\text{Current_Basic_Pay}/100) * 0.1$
- GROSS_PAY = `Current_Basic_Pay + DA + HRA + PF`
- NET_PAY = `GROSS_PAY - PF`
- Step 17 Display Payment Details [Name, ID, Address, Mail ID, Mobile Number, BASIC PAY, DA, HRA,PF,GROSS_PAY, NET_PAY].
- Step 18 Stop Processing

Coding

Employee.java

```
package com.raja.oopslab.employee;
import java.util.Calendar;
import java.util.GregorianCalendar;
import java.util.Scanner;

package com.raja.oopslab.employee;

import java.util.Calendar;
import java.util.GregorianCalendar;
import java.util.Scanner;

class Employee {
    String emp_name;
    String emp_id;
    String emp_address;
    String emp_mail_id;
    String emp_mobile_no;
    int basic_pay;
    int per_day_pay;
    int current_basic_pay;
    int da, hra, pf, gross_pay;
    int net_pay;
    int no_of_days_in_current_month;
    int no_of_days_worked;
    Calendar cal;
    Scanner input;

    Employee() {
        input = new Scanner(System.in);
        cal = new GregorianCalendar();
        no_of_days_in_current_month =
cal.getActualMaximum(Calendar.DAY_OF_MONTH);
        getUserBasicDetails();
    }

    public void generatePaySlip() {
        this.da = (this.current_basic_pay / 100) * 97;
        this.hra = (this.current_basic_pay / 100) * 12;
        this.pf = (int) ((this.current_basic_pay / 100) * 0.1);
        this.gross_pay = this.current_basic_pay + da + hra + pf;
        this.net_pay = gross_pay - pf;
    }

    public void displayPaySlip() {
        System.out.println("Name: " + this.emp_name);
        System.out.println("ID: " + this.emp_id);
        System.out.println("Address: " + this.emp_address);
        System.out.println("MailID: " + this.emp_mail_id);
    }
}
```

```

        System.out.println("Mobile No: " + this.emp_mobile_no);
        System.out.println("\nEarnings");
        System.out.println("-----");
        System.out.println("BASIC Pay: " + current_basic_pay + " Rs");
        System.out.println("DA : " + da + " Rs");
        System.out.println("HRA : " + hra + " Rs");
        System.out.println("\nDeductions");
        System.out.println("-----");
        System.out.println("PF : " + pf + " Rs");
        System.out.println("GROSS Pay: " + gross_pay + " Rs");
        System.out.println("NET Pay: " + net_pay + " Rs");
    }

    public void getUserBasicDetails() {
        System.out.println("Enter Details");
        System.out.println("Name: ");
        this.emp_name = input.next();
        System.out.println("ID: ");
        this.emp_id = input.next();
        System.out.println("Address: ");
        this.emp_address = input.next();
        System.out.println("MailID: ");
        this.emp_mail_id = input.next();
        System.out.println("Mobile No:");
        this.emp_mobile_no = input.next();
    }

    public void computeCurrentBasicPay(String empType) {
        this.per_day_pay = this.basic_pay / no_of_days_in_current_month;
        System.out.println("\nBasic Pay of " + empType + " " + this.basic_pay + " for "
            + this.no_of_days_in_current_month + " days");
        System.out.println("This month this " + empType + " gets " + this.per_day_pay + "
INR as basic pay per day");
        System.out.println("Enter no.of days worked by " + empType + " including CL, WH,
FH and excluding LWP:");
        this.no_of_days_worked = input.nextInt();
        if (no_of_days_worked <= no_of_days_in_current_month) {
            this.current_basic_pay = this.per_day_pay * no_of_days_worked;
            System.out.println("Programmer");
            System.out.println("-----");
            generatePaySlip();
        } else {
            System.out.println("Sorry Please Enter Valid Days");
        }
    }

    protected void finalize() {
        input.close();
        System.exit(0);
    }
}

```

Programmer.java

```
package com.raja.oopslab.employee;

public class Programmer extends Employee {

    public Programmer() {
        super();
        computeProgrammerPay();
    }

    public void computeProgrammerPay() {
        System.out.println("Enter Basic pay of Programmer [enter -1 for Default [BP = 30000]]:");
        this.basic_pay = input.nextInt();
        if (this.basic_pay == -1) {
            this.basic_pay = 30000;
            System.out.println("Default Pay Taken");
        }
        computeCurrentBasicPay("Programmer");
        generatePaySlip();
        displayPaySlip();
    }
}
```

Assistant Professor.java

```
package com.raja.oopslab.employee;

public class AssistantProfessor extends Employee {
    public AssistantProfessor() {
        super();
        computeAssistantProfessorPay();
    }

    public void computeAssistantProfessorPay() {
        System.out.println("Enter Basic pay of AssistantProfessor [enter -1 for Default [BP
= 25000]]:");
        this.basic_pay = input.nextInt();
        if (this.basic_pay == -1) {
            this.basic_pay = 25000;
            System.out.println("Default Pay Taken");
        }
        computeCurrentBasicPay("AssistantProfessor");
        generatePaySlip();
        displayPaySlip();
    }
}
```

Associate Professor

```
package com.raja.oopslab.employee;

public class AssociateProfessor extends Employee {
    public AssociateProfessor() {
        super();
        computeAssociateProfessorPay();
    }

    public void computeAssociateProfessorPay() {
        System.out.println("Enter Basic pay of AssociateProfessor [enter -1 for Default [BP
= 40000]]:");
        this.basic_pay = input.nextInt();
        if (this.basic_pay == -1) {
            this.basic_pay = 40000;
            System.out.println("Default Pay Taken");
        }

        computeCurrentBasicPay("AssociateProfessor");
        generatePaySlip();
        displayPaySlip();
    }
}
```

Professor

```
package com.raja.oopslab.employee;

public class Professor extends Employee {
    public Professor() {
        super();
        computeProfessorPay();
    }

    public void computeProfessorPay() {
        System.out.println("Enter Basic pay of Professor [enter -1 for Default [BP =
70000]]:");
        this.basic_pay = input.nextInt();
        if (this.basic_pay == -1) {
            this.basic_pay = 70000;
            System.out.println("Default Pay Taken");
        }
        computeCurrentBasicPay("Professor");
        generatePaySlip();
        displayPaySlip();
    }
}
```



```
default:  
    System.out.println("Enter valid choice !!!");  
    break;  
}
```

```
}
```

```
}
```

```
}
```

Output:

Choices

```
Markers Properties Servers Data Source Explorer Snippets Console
Main (11) [Java Application] /usr/lib/jvm/java-8-openjdk-amd64/bin/java (01-Jun-2018, 11:14:46 AM)

Employee Payroll System
*****

1. Programmer
2. Assistant Professor
3. Associate Professor
4. Professor
5. Exit

Enter Your Choice
```

Basic Details

```
Markers Properties Servers Data Source Explorer Snippets Console
Main (11) [Java Application] /usr/lib/jvm/java-8-openjdk-amd64/bin/java (01-Jun-2018, 11:14:46 AM)
Enter Your Choice
1
Programmer Selected
Enter Details
Name:
Raja
ID:
987654
Address:
Coimbatore
MailID:
raja@gmail.com
Mobile No:
74874928921
Enter Basic pay of Programmer [enter -1 for Default [BP = 30000]]:
40000
```

Programmer

```
Markers Properties Servers Data Source Explorer Snippets Console
Main (11) [Java Application] /usr/lib/jvm/java-8-openjdk-amd64/bin/java (01-Jun-2018, 11:14:46 AM)

Basic Pay of Programmer 40000 for 30 days
This month this Programmer gets 1333 INR as basic pay per day
Enter no.of days worked by Programmer including CL, WH, FH and excluding LWP:
28
Programmer
-----
Name: Raja
ID: 987654
Address: Coimbatore
MailID: raja@gmail.com
Mobile No: 74874928921

Earnings
-----
BASIC Pay: 37324 Rs
DA : 36181 Rs
HRA : 4476 Rs

Deductions
-----
PF : 37 Rs
GROSS Pay: 78018 Rs
NET Pay: 77981 Rs
```

Assistant Professor

```
Markers Properties Servers Data Source Explorer Snippets Console
Main (11) [Java Application] /usr/lib/jvm/java-8-openjdk-amd64/bin/java (01-Jun-2018, 11:21:34 AM)
Enter Basic pay of AssistantProfessor [enter -1 for Default [BP = 25000]]:
-1
Default Pay Taken

Basic Pay of AssistantProfessor 25000 for 30 days
This month this AssistantProfessor gets 833 INR as basic pay per day
Enter no.of days worked by AssistantProfessor including CL, WH, FH and excluding LWP:
29
Pay Slip
-----
Name: Selva
ID: 099893
Address: Saravanapatti
MailID: selva@gmail.com
Mobile No: 989823892

Earnings
-----
BASIC Pay: 24157 Rs
DA : 23377 Rs
HRA : 2892 Rs

Deductions
-----
PF : 24 Rs
GROSS Pay: 50450 Rs
NET Pay: 50426 Rs
```

Associate Professor

```
Markers Properties Servers Data Source Explorer Snippets Console
Main (11) [Java Application] /usr/lib/jvm/java-8-openjdk-amd64/bin/java (01-Jun-2018, 11:21:34 AM)
Enter Basic pay of AssociateProfessor [enter -1 for Default [BP = 40000]]:
-1
Default Pay Taken

Basic Pay of AssociateProfessor 40000 for 30 days
This month this AssociateProfessor gets 1333 INR as basic pay per day
Enter no.of days worked by AssociateProfessor including CL, WH, FH and excluding LWP:
30
Pay Slip
-----
Name: Mani
ID: 983982
Address: Sulur
MailID: mani@gmail.com
Mobile No: 9389892208

Earnings
-----
BASIC Pay: 39990 Rs
DA : 38703 Rs
HRA : 4788 Rs
|
Deductions
-----
PF : 39 Rs
GROSS Pay: 83520 Rs
NET Pay: 83481 Rs
```

Professor

```
Markers Properties Servers Data Source Explorer Snippets Console
Main (11) [Java Application] /usr/lib/jvm/java-8-openjdk-amd64/bin/java (01-Jun-2018, 11:21:34 AM)
Enter Basic pay of Professor [enter -1 for Default [BP = 70000]]:
-1
Default Pay Taken

Basic Pay of Professor 70000 for 30 days
This month this Professor gets 2333 INR as basic pay per day
Enter no.of days worked by Professor including CL, WH, FH and excluding LWP:
27
Pay Slip
-----
Name: Anvar
ID: 847479
Address: Erode
MailID: anvar@gmail.com
Mobile No: 9379212080

Earnings
-----
BASIC Pay: 62991 Rs
DA : 61013 Rs
HRA : 7548 Rs

Deductions
-----
PF : 62 Rs
GROSS Pay: 131614 Rs
NET Pay: 131552 Rs
Thank You !!!
```

Result:

The java console application for employee payroll system was developed and tested successfully.