

## 1.Problem Statement

Demonstrate the usage of super and this keyword with a constructor,methods,variables.

## 2.Source Code

```
//Demonstrate the use of Super and this keyword with a constructor,methods,variables
package mypackage;
import java.util.Scanner;
```

```
//Defining Superclass Employee
```

```
class Employee
{
    //Data members of Employee class
    int rent;
    int medical;
    int pf;
    int salary;
```

```
//Defining constructor for Parent class Employee
```

```
Employee()
{
    rent=1000;
    medical=300;
    pf=200;
}
```

```
//Defining parameterized constructor of Parent class Employee
```

```
Employee(int rent,int medical,int pf)
{
    this.rent=rent;
    this.medical=medical;
    this.pf=pf;
}
```

```
//Defining method to calculate salary
```

```
void setSalary()
{
    salary=rent+medical+pf;
}
```

```
// Method to return the salary
```

```
int getSalary()
{
    return salary;
}
```

```
}
```

```
//Child class inherits the data members of parent class Employee
```

```
class Manager extends Employee
```

```
{
```

```
    //Constructor definition for child class Manager
```

```
    Manager(int r,int m,int p)
```

```
    {
```

```
        rent=r;
```

```
        medical=m;
```

```

        pf=p;
    }
}

//Child class to inherit the data members of Parent class Employee
class Developer extends Employee
{
    Developer(int r,int m, int p)
    {
        //To call the parameterized constructor of
parent class Employee
        super(r,m,p);
    }
}

public class DemoSuper {
    //Defining Main method
    public static void main(String[] args) {
        int sal;
        System.out.println("Salary details of Manager");
        System.out.println();
        Scanner sc=new Scanner(System.in);

        //To enter the salary details of manager
        System.out.println("Enter the House rent amount:");
        int ramt=sc.nextInt();
        System.out.println("Enter Medical allowance amount:");
        int mamt=sc.nextInt();
        System.out.println("Enter Provident fund amount:");
        int pamt=sc.nextInt();

        //Creating object for Manager
        Manager e1=new Manager(ramt,mamt,pamt);
        //Calling setSalary method on Manager object
        e1.setSalary();
        //Calling getSalary method on Manager object
        sal=e1.getSalary();
        //Displaying total salary of manager
        System.out.println();
        System.out.println("Total Salary of manager is :"+sal);

        System.out.println();

        System.out.println("Salary details of Developer:");
        System.out.println();
        //To enter the salary details of Developer
        System.out.println("Enter the House rent amount:");
        ramt=sc.nextInt();
        System.out.println("Enter Medical allowance amount:");
        mamt=sc.nextInt();
        System.out.println("Enter Provident fund amount:");
        pamt=sc.nextInt();
    }
}

```

```

        //Creating object for Developer
        Developer e2=new Developer(ramt,mamt,pamt);

        //Calling setSalary method on developer object
        e2.setSalary();
        //Calling getSalary method on developer object
        sal=e2.getSalary();

        System.out.println("Total Salary of developer is :"+sal);
    }
}
} //End of Program

```

### 3.Output

```

<terminated> DemoSuper [Java Application] C:\Java\jre-10\bin\javaw.exe (Nov 22, 2018, 5:36:21 PM)
Salary details of Manager

Enter the House rent amount:
10000
Enter Medical allowance amount:
5000
Enter Provident fund amount:
3000

Total Salary of manager is :18000

Salary details of Developer:

Enter the House rent amount:
6000
Enter Medical allowance amount:
2000
Enter Provident fund amount:
1000
Total Salary of developer is :9000

```

### 4.Flow Information

The program calculates and displays the total salary of a manager and developer. At first Employee super class is created with data members of salary details. Also, setter method is defined to calculate the salary as sum of rent amount, medical allowance amount and pf amount. In the same class, the getter method is defined to return the value of calculated salary.

Two child classes Manager, Developer are inherited from parent class Employee and the respective class constructors are called implicitly on creation of objects. User is asked to enter the salary details of manager and developer. The SetSalary and getSalary methods are called with help of the objects created for child classes. 'Super' keyword is used in child class Developer to call the Employee super class constructor. 'This' keyword is used in parameterized constructor of Employee class.

Finally, the salary of Manager and Developer are displayed as output.