**Constructor :** TASK 1

package day2;

class Employee{

int empid;

String name;

float sal;

public Employee(int empid,String name,float sal)

{

this.empid=empid;

this.name=name;

this.sal=sal;

}

void display()

{

System.***out***.println(empid);

System.***out***.println(name);

System.***out***.println(sal);

}

}

public class task1 {

public static void main(String args[])

{

Employee emp1 = new Employee(001,"hari",13000);

emp1.display();

}

}

TASK 2:

Calculate salary of the employee.

package day2;

class Employee2{

int empid;

String name;

float sal;

float salary;

public Employee2(int empid,String name,float sal)

{

this.empid=empid;

this.name=name;

this.sal=sal;

}

void display()

{

System.***out***.println(empid);

System.***out***.println(name);

System.***out***.println(sal);

}

void salary()

{

salary=(float) (this.sal+(this.sal\*0.05f));

}

}

public class task2 {

public static void main(String args[])

{

Employee2 emp1 = new Employee2(001,"hari",13000);

emp1.display();

emp1.salary();

System.***out***.println(emp1.salary);

}

}

TASK 3 :

CONSTRUCTOR OVERLLOADING

package day2;

//constructor over loading..

class Employee3{

int empid;

String name;

float sal;

float salary;

public Employee3(int empid,String name,float sal)

{

this.empid=empid;

this.name=name;

this.sal=sal;

}

public Employee3() {

System.***out***.println("default constructor");

}

public Employee3(int empid,float sal) {

this.name=name;

this.empid=empid;

this.sal=sal;

System.***out***.println("second constructor");

}

public Employee3(String name,int empid,float sal) {

this.name=name;

this.empid=empid;

this.sal=sal;

System.***out***.println("third constructor");

}

void display()

{

System.***out***.println(empid);

System.***out***.println(name);

System.***out***.println(sal);

}

float salary()

{

this.sal=(this.sal+(this.sal\*0.05f));

return this.sal;

}

}

public class task3 {

public static void main(String args[])

{

Employee3 emp1 = new Employee3(55,"hari",13000);

emp1.display();

Employee3 emp2= new Employee3(); //default constructor

Employee3 emp3 = new Employee3(22,2222); //2 argu constructor

Employee3 emp4 = new Employee3("haran",55,13000);// third const is work --> using const overloading

Employee3 emp5 = new Employee3("haran",'H',13000);//implicit conversion --> (ch-int)

emp4.display();

emp5.display();

//System.out.println(emp1.salary());

}

}

Constructor chaining:

package day2;

//constructor chaining

class Employee4{

int empid;

String name;

float sal;

float salary;

String email;

static int *emailcount*=0;

public Employee4() {

System.***out***.println("default constructor");

*emailcount*+=1;

}

public Employee4(int empid,String name) {

System.***out***.println("second constructor");

// emailcount+=1; avoid chain to default const

}

public Employee4(int empid,String name,float sal)

{

this();

this(empid,name);

// emailcount+=1; avoid chain to default const

}

public Employee4(String name,int empid,float sal) { /\* order change \*/

System.***out***.println("third constructor");

// emailcount+=1; avoid chain to default const

}

void display()

{

System.***out***.println(empid);

System.***out***.println(name);

System.***out***.println(sal);

}

float salary()

{

this.sal=(this.sal+(this.sal\*0.05f));

return this.sal;

}

}

public class task4 {

public static void main(String args[])

{

Employee4 emp1 = new Employee4(55,"hari",13000);

Employee4 emp2= new Employee4();

System.***out***.println(Employee4.*emailcount*);

Employee4 emp3 = new Employee4(22,"kumar");

Employee4 emp4 = new Employee4("haran",55,13000);

Employee4 emp5 = new Employee4("haran",'H',13000);

emp4.display();

emp5.display();

System.***out***.println(Employee4.*emailcount*);

}

}

Get and set method :

package day2;

class Emp{

private int empid;

String name;

float sal;

public Emp(int empid,String name,float sal)

{

this.empid=empid;

this.name=name;

this.sal=sal;

}

void display()

{

System.***out***.println(empid);

System.***out***.println(name);

System.***out***.println(sal);

}

public int getEmpId() { /\* use the empid in main functionwe are putting the public method

--> through this we access the private empid in main function

\*/

return this.empid;

}

public void setEmpId(int empid)

{

this.empid=empid;

}

}

public class task5 {

public static void main(String args[])

{

Emp emp1 = new Emp(55,"hari",13000);

emp1.setEmpId(57); //cal the setEmpId with 57

System.***out***.println(emp1.getEmpId());// get the public empid to using the public getempId method

}

}

ACCESS SPECIFIER :

PUBLIC , PRIVATE , PROTECTED,DEFAULT :

package day2;

// ACCESS SPECIFIER

class second

{

public int num1=10;

private int num2=11;

protected int num3=199;

public int access()

{

return num2;

}

}

public class task6 {

public static void main(String args[])

{

task6 t = new task6();

second s = new second();

//System.out.println(t.num1); num1 cannot be resolved or is not a field --> num1 is public but can't acess by the outter class

System.***out***.println(s.num1);

//System.out.println(second.num1); --> num1 not a static

//System.out.println(s.num2); --private

//System.out.println(s.access()); --> create a public methos in inside the second class to access the private variable

System.***out***.println(s.num3);

}

}

Extends class access the protected var :

package day2;

class first

{

public int num1=10;

private int num2=11;

protected int num3=199;

public int access()

{

return num2;

}

}

class sec extends first{

void display()

{

}

}

public class Task7 {

public static void main(String args[])

{

first s = new first();

sec ss = new sec();

System.***out***.println(ss.num3);

}

}

INHERITANCE

package day2;

class Employeee

{

int empid;

String name;

float sal;

public Employeee(int empid,String name,float sal)

{

this.empid=empid;

this.name=name;

this.sal=sal;

}

void display()

{

System.***out***.println(empid);

System.***out***.println(name);

System.***out***.println(sal);

}

}

class Developer extends Employee{

}

public class task9 {

public static void main(String args[])

{

Developer dev = new Developer();

}

}

Op :

ERROR IS OCCUR because the developer class is default constructor but it inherite the parent class …so error is occur….

>>Implicit super constructor Employee() is undefined for default constructor. Must define an explicit constructor

>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>

Employee default is invoke for developer object

Parent class invoke before the child class

package day2;

class Employeee

{

int empid;

String name;

float sal;

public Employeee() {

System.***out***.println("default constructor");

}

public Employeee(int empid,String name,float sal)

{

this.empid=empid;

this.name=name;

this.sal=sal;

}

void display()

{

System.***out***.println(empid);

System.***out***.println(name);

System.***out***.println(sal);

}

}

class Developer extends Employeee{

public Developer()

{

System.***out***.println("developer obj is created");

}

}

public class task9 {

public static void main(String args[])

{

Developer dev = new Developer();

}

}