28-01-2013 | MONDAY

CHAPTER

Constructors

P47

1)

```
class A
{
  int i=10;
  A()
  {
    Sop("running class A constructor");
  }
}

class Run1
{
   psvm( )
   {
    Sop("program starts...");
    // refer i member of class A
    A a1 =new A();
    Sop("i="+a1.i);
    Sop("program ends...");
   }
```

O/p:

```
Program starts...
running class A constructor
i=10
Program ends...
```

Syntax of Constructor:-

```
ConstructorName()
{
  Constructor body
}
```

1*) A constructor is a block just like method which is invoked at the time of Object Creation.

2*) The constructor should be written with above Syntax.

3*) The name of constructor should be same as Class Name.

4*) The constructor body should not have return statement

5*) The constructor should not be declared with any return type.

6*) Whenever a constructor is invoked, the body of constructor get executed.

7*) A constructor is developed to initialize Object variables at the time of Object Creation.

8*) Every java class should have a constructor, if we don't develop a constructor compiler writes a constructor known as default Constructor

2) P48

```
class B
{
    double d=3.4;
    B()
    {
        Sop("running Class B constructor");
        d=12.45;
    }
}

class Run2
{
    psvm()
    {
        Sop("Program starts...");
        B b1 = new B();
        Sop("d="+b1.d);
        Sop("----------");
        B b2 = new B();
        Sop("d="+b2.d);
        Sop("Program ends...");
    }
}
```

O/p :-

```
Program starts...
running class B constructor
d=12.45
----------
running class B constructor
d=12.45
Program ends...
```

9.* Default Constructor doesn't have any arguments
The body of the default will have mandatory statements.

10.* While developing a constructor we can pass a value as a argument.

11.* A constructor which is defined with arguments are known parameterized constructor (or) argument constructor.

12.* Developing more then one constructor in a Class is known as constructor overloading (or) Overloaded Constructor.

In order to develop this, following critirea should be fulfilled.

12.1.* The argument list/parameter should differ interms of datatype.

12.2.* The arguments should differ interms of number of args.

12.3.* The argument should differ with respect to position

13.* In order to develop Overloaded Constructor any of the above should be fulfilled.

3)

```
Class C
{
  double d=3.4;
   C (double a)
   { System.out.p("running C() constructor");
     d=a;
   }
}

class Run3
{
  psvm ( )
  {
   Sop ("Program starts...");
   C c1 = new C(2.3);
   Sop ("d="+c1.d);
   Sop ("----------");
   C c2 = new C(34.21);
   Sop ("d="+c2.d);
   Sop ("Program ends...");
  }
}
```

O/P:

```
Program starts...
running C() constructor
d=2.3
----------
running C() constructor
d=34.21
Program ends...
```

a Class is to be created with different implementation or data we go for constructor Overloading.

15*) If we develop any type of constructor inside the Class, then compiler will not write default Constructor

" Compiler writes default Constructor only if Class doesn't have any type of Constructor "

```
class D
{
  int i;
  double d;
  //overloaded constructor
  D(int a)
  {
    Sop("running D(int) constructor");
    i=a;
  }
  D(double b)
  {
    Sop("running D(double) constructor");
    d=b;
  }
  D(int a, double b)
  {
    Sop("running D(int, double) constructor");
    i=a;
    d=b;
  }
  void println()
  {
    Sop("i="+i);
    Sop("d="+d);
  }
}
Class Run4
{
  psvm()
  {
    Sop("Program starts...");
    D d1 = new D(12);
    d1.println();
    Sop("----------");
    D d2 = new D(6.32);
    d2.println();
    Sop("----------");
    D d3 = new D(78, 34.53);
    d3.println();
    Sop("Program ends...");
  }
```

continued program...

O/P :-

```
Program starts...
running D(int) constructor
i=12
d=0.0
----------
running D(double) constructor
i=0
d=6.32
----------
running D(int, double) constructor
i=78
d=34.53
Program ends.
```

29-01-2013 | Tuesday

5) P51

```
class E
{
   { Sop("running non-static block");
   }
   E()
   { Sop("running E() constructor");
   }
}

class Run6
{
  psvm()
  {
   Sop("Program starts...");
   E e1 = new E();
   Sop("------------");
   E e2 = new E();
   Sop("Program ends...");
  }
}
```

O/P:

```
Program starts...
running non-static block
running E() constructor
------------
running non-static block
running E() constructor
Program ends...
```

1*) If a Class refers member of another Class, then first static block of referred Class will be executed.

If the current/running Class contains static block, then the static block of the running Class will be executed first, then static block of referred Class will be executed.

2*) If in a running Class a reference is made through Object, then the static block of the running Class gets executed first, then static block of referred Class will be executed first and then non-static block of referred Class will be executed and then constructor body will be executed.

3*) A constructor invokes a non-static block whenever an Object is created.

6) P52

```
class E
{
  static
  {
    Sop("running static block of class E");
  }
  {
    Sop("running non-static block ");
  }
  E()
  {
    Sop ("running E() constructor");
  }
}
class Run6
{
  static
  {
    Sop("running static block of Class Run6");
  }
  psvm ( )
  {
    Sop("Program starts...");
    E e1 = new E();
    Sop("-----------");
    E e2 = new E();
    Sop("Program ends...");
  }
}
```

O/p:

```
Running static block of Class Run6
Program starts...
Running static block of Class E
Running non-static block
Running E() constructor
-----
running non-static block
running E() constructor
Program ends...
```

this keyword

P53

1)
```
class F
{
  int i;
  double d;
  F(int i, double d)
  {
   Sop("running F() constructor...");
   this.i = i;
// global i = local i
   this.d = d;
  }
  void println()
  {
   System.Op("i=" + i);
   Sop("d=" + d);
  }
}
Class Run7
{
   psvm( )
   {
   Sop("Program starts...");
   F f1 = new F(123, 78.56);
   f1.print()
   Sop("Program ends...");
   }
}
```

O/P:
```
Program starts...
running F() constructor...
i=123
d=78.56
Program ends...
```

1*) 'this' keyword refers to current Class instance, using 'this' keyword we can refers the non-static member of the current Class with the Class. 'this' keyword has to be used in constructor (or) inside non-static context.

2*) A constructor of a Class can Call another constructor of Same class, this can be done by using "this()" statement. Using 'this()' you can call any constructor of current Class.

Recursive constructor Call is not allowed in java.

3*) 'this()' must be the first statement in constructor body.

P54

2)

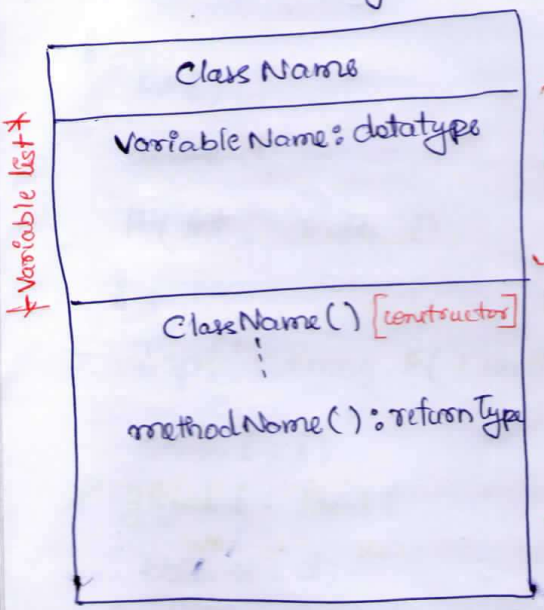
```
class G
{
  G()
  {
    Sop("running default() constructor");
  }
  G(int i)
  {
    this();
    Sop("running G(int) constructor");
  }
  G(double d)
  {
    this(23); // Statement
    Sop("running G(double) constructor");
  }
}

class Run8
{
  psvm()
  {
    Sop("Program starts...");
    G g1 = new G(54.34); //double type Constructor
    Sop("Program ends...");
  }
}
```

O/p: Program starts...
running default() constructor
running G(int) constructor
running G(double) constructor
Program ends...

30-01-2013 | WEDNESDAY

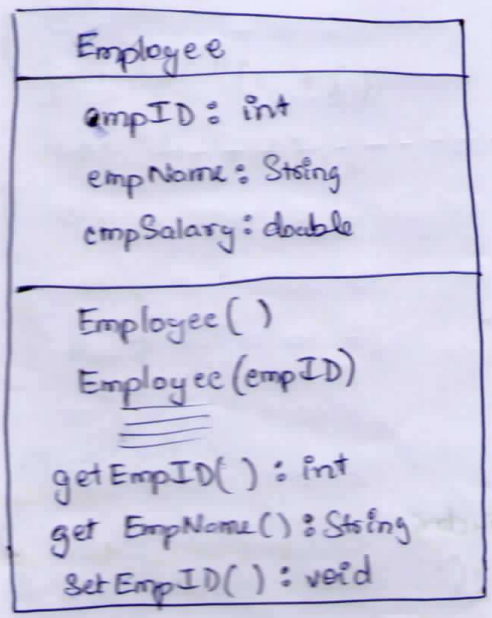
Class Diagram



1*) Class Diagram is pictorial representation of java Class, which describes the behaviour and properties of the Class.

2*) Class Diag are used while designing the relationship of Class.

Ex:



Assign: Develop a program for above Class Diagram

```
Class Employee
{
  int empID;
  String empName;
  double empSalary;

  Employee()
  { Sop("Employee List of Jspiders");
  }
  Employee(int empID)
  { Sop("EmpID="+empID);
  } this.empID = empID;
  Employee(int empID, double empSalary)
  { this.empID = empID;
    this.empSalary = empSalary;
  }
  Employee(String empName)
  { this.empName = empName;
  }
  void print()
  { Sop("Employee "+empName+"Emp
      ID "+empID+"with Salary"+
      empSalary);
  }
```

String