**Object Reference:**

* An object of a class can be assigned an instance of the same class, the object becomes an object reference.
* Object reference are operational.
* Typically, when an instance of a class is sent as an argument into a method of a Java-object, the parameter to receive the argument becomes an object-reference.

void write( Pen P) //This parameter is the object reference of

{ object P1.

\_\_\_\_\_\_\_\_

}

Person tom, mike;

Pen p1,p2;

Tom = new Person();

tom.talk();

tom.walk();

p1 = new Pen();

tom.write(p1);

Pen p1, p2;

P1 = new Pen();

P2 = P1; //Object referencing. Both objects point at the same elements.

**Constructors:**

* These are special methods, which should have the same name as the class itself.
* **It can have parameters but no RETURN TYPE, not even void.**
* A constructor with no parameter is called default constructor and one with parameter is called parameterized constructor.
* If a constructor is not defined for a class the compiler provides a default constructor devoid of any code.
* A constructor is meant to instantiate an object into an instance bringing it to some usable initial state.

Public class Pen

{

int cap; //Instance Data.

String name,clr;

Boolean open;

Public Pen() // Default constructor.

{

cap = 20;

open = false;

name = “Reynolds”;

clr = “Black”;

}

public Pen(string c) //Parameterized constructor. //Local Data

{

cap = 20;

open = false;

name=”Reynolds”;

clr = c;

}

}

PROGRAM:

Pen p1,p2;

p1 = new Pen();

p2 = new Pen(“Red”);

**Instance Data and Local Data:**

|  |  |
| --- | --- |
| Instance Data | Local Data |
| 1. Declared OUTSIDE of all the constructors and methods. 2. Doesn’t require forward declaration. 3. A separate copy per instance basis. 4. Contributes to objects directly. 5. Initialized automatically upon construction. 6. Default values: Primitives has natural known values while objects of class have null values | 1. Declared inside the constructors and methods. 2. Local data requires forward declaration. 3. Created when a constructor or a method is invoked. 4. Contributes indirectly upon assignment. 5. Has to be explicitly initiated. 6. Default values need to be given by the programmer. |

**SCOPE AND VISIBILITY:**

* Scope means for how much time a data exists. Visibility means which all functions can use it.
* Instance Data:
  + Scope of instance data is as long as the instance is there.
  + Visibility of instance data is the complete class with **one exception**. (To be covered later).
* Local Data:
  + Scope: As long as inside the constructor or method.
  + Visibility: Only statements inside of the corresponding constructor/method.

**SHADOWING:**

* When names of local and instance data are same shadowing occurs.
* ‘this’ operator is used to differentiate between local and instance data having same name.