Todays Topic

- OOPS
  - Objects
  - Object Reference
  - Object Initialization

OOPS – Object Oriented Programming System

Java is an object oriented language.

It provides us a programming environment where we can create objects and can perform operations on them.

Object

Components of Objects

- **Data**
  - PenColor
  - InkColor
  - InkQty
  - Length
  - Radius
  - Company
  - Price
- **Functions**
  - Fill( )
  - Write( )
  - Throw( )

Class :

A class is a user defined data type where we can group data and its related functions together.

```
class className
{
      Data Members
      . . . . .
      . . . . .
      Member Functions
      . . . . .
      . . . . .
}
```

Data members:

```
[AccessSpecifier] [Modifier] datatype memberName [=value] ;
```

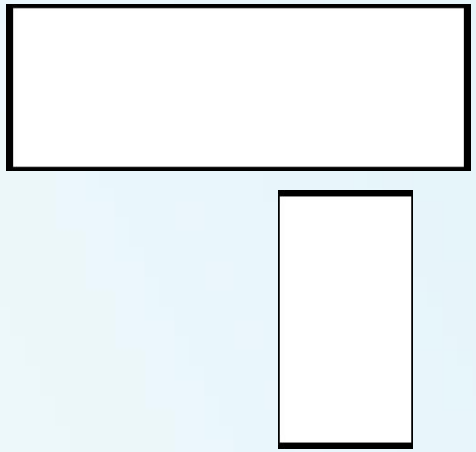
Member functions :

```
[AccessSpecifier] [Modifier]  returntype fName ( datatype arg1, . . )
    {
    Statements.....
    ..........................
    return value;
    }
```

```
class className
{
    Data Members
    . . . . .
    . . . . .
    Member Functions
    . . . . .
    . . . . .
}
```

For ex:

```
class Rectangle
{
int length;
int breadth;
void area( )
      {
      int a=length*breadth;
      System.out.println('Area is "+a);
      }
void perimeter( )
      {
      int p=2*(length+breadth);
      System.out.println('Perimeter  is "+p);
      }
}
```



Data members
- Length
- Breadth

Member Functions
- Area()
- Perimeter()

[AccessSpecifier] [Modifier] datatype memberName [=value] ;

```
[AccessSpecifier] [Modifier]  returntype fName(datatype arg1, . . )
      {
      Statements…..
      ……................
      return value;
      }
```

Objects

- An Object is instance of a class.
- Objects in java can be dynamically created by using operator **new** .

- Syntax: **ObjectReference = new ClassName( ) ;**

- For ex:

  a = new Rectangle( ) ;

  b = new Rectangle( ) ;

Where Rectangle is a classname and a,b are object references.

Object Reference

- It is a variable in which we can store ID of an Object.
- Each Object has an unique ID.
- An object reference can be defined just like variables.
- Syntax: **ClassName ReferenceList**;

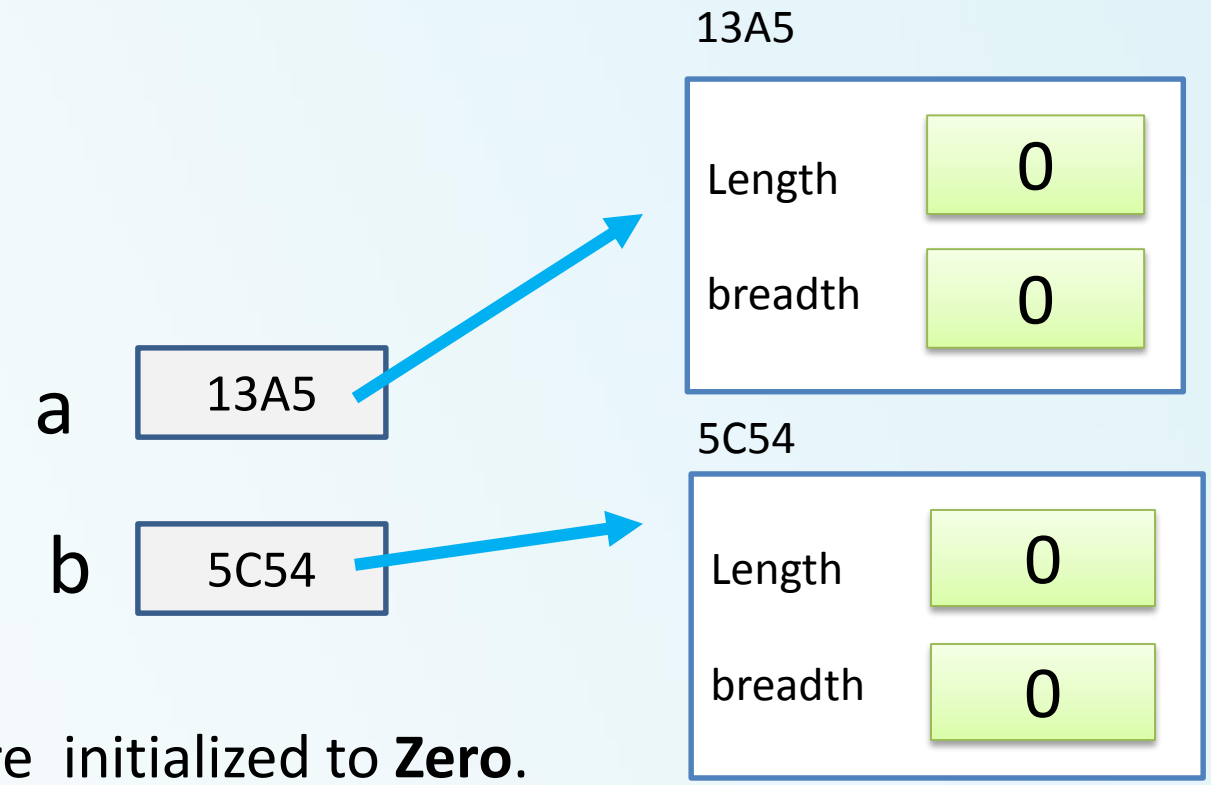
- For ex:

Rectangle a , b , c ;

Due to this 3 object references will be created in which we can store id of objects.

Object Initialization

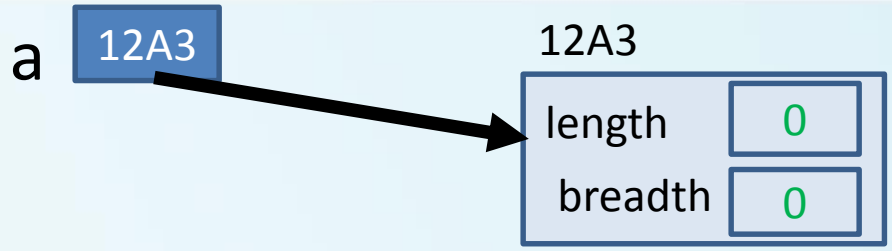
- Whenever an object is created space is reserved for objects data members.
- For ex:
  - Rectangle a, b ;
  - a = new Rectangle( );
  - b = new Rectangle( );



Note:

1. Numeric Data members are initialized to **Zero**.
2. Boolean types are initialized to **false**.
3. Object References are Initialized to **null**.

```
class rectangle
{
    int length , breadth;
    void area()
        {
        int a = length*breadth;
        System.out.println("Area is "+a)
        }
    void perimeter()
        {
        int p = 2*( length+breadth );
        System.out.println("perimerter is"+p);
        }
}
```

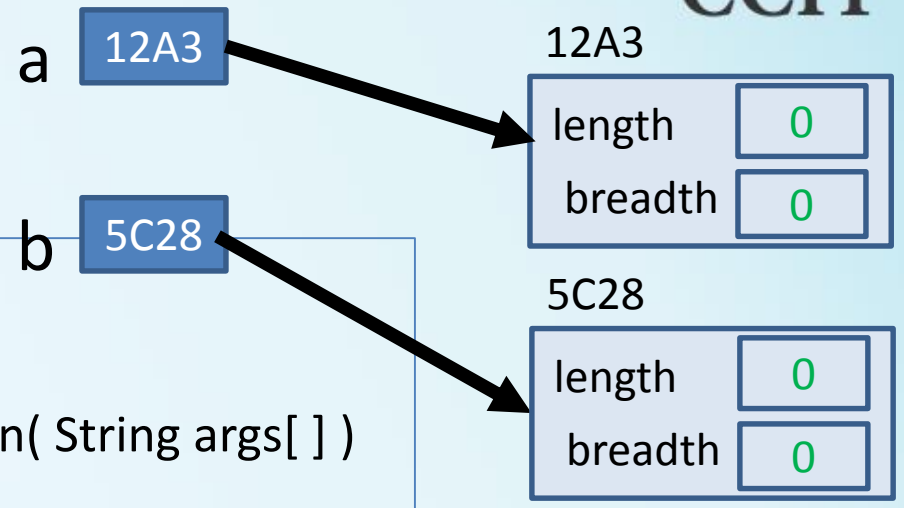



```
class demo
{
public static void main( String args[ ] )
  {
  rectangle a ;
  a = new rectangle( ) ;
  System.out.println(a) ;
 }
}
```

Output :

rectangle@12A3

```
class rectangle
{
    int length , breadth;
    void area()
        {
        int a = length*breadth;
        System.out.println("Area is "+a)
        }
    void perimeter()
        {
        int p = 2*( length+breadth );
        System.out.println("perimerter is"+p);
        }
}
```



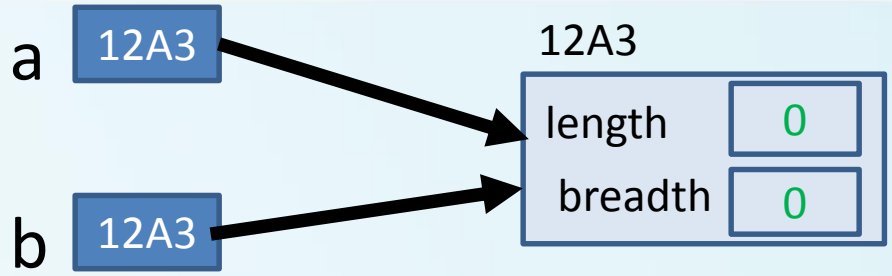

```
class demo
{
public static void main( String args[ ] )
  {
  rectangle a , b ;
  a = new rectangle( ) ;
  b = new rectangle( ) ;
  System.out.println( a ) ;
  System.out.println( b ) ;
  }
}
```

Output :

rectangle@12A3

rectangle@5C28

```
class rectangle
{
    int length , breadth;
    void area()
        {
        int a = length*breadth;
        System.out.println("Area is "+a)
        }
    void perimeter()
        {
        int p = 2*( length+breadth );
        System.out.println("perimerter is"+p);
        }
}
```




```
class demo
{
public static void main( String args[ ] )
  {
  rectangle a , b ;
  a = new rectangle( ) ;
  b = a ;
  System.out.println( a ) ;
  System.out.println( b ) ;
  }
}
```

Output :

```
rectangle@12A3
rectangle@12A3
```

Todays Topic End