Todays Topic

- OOPS
- Member Access Operator

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.

class:

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

Data members:

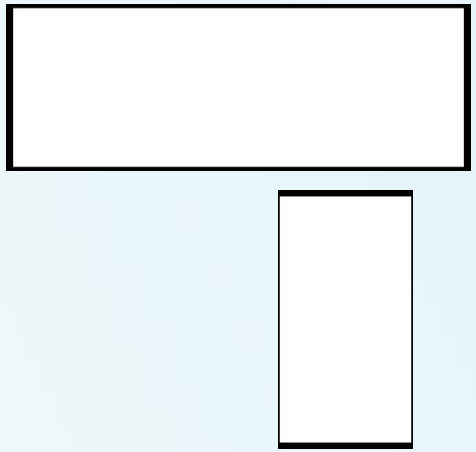
```
[AccessSpecifier] [Modifier] datatype memberName [=value] ;
```

Member functions :

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

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( );`

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 ;`

- Member Access Operator(.)
- Members of an object can be access by using member access operator

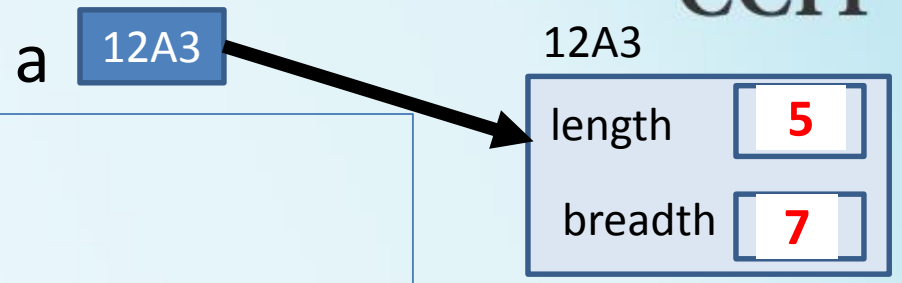
- Syntax: **ObjectName.MemberName**

- **For ex : suppose a is an object of class rectangle**
  - **a.length**
  - **a.breadth**
  - **a.area( )**
  - **a.perimeter( )**

//program to create an rectangle type of object of size 5 x 7 and display its length breadth area and perimeter

```
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);
        }
}
```

5 X 7




```
class demo
{
public static void main( String args[ ] )
  {
  rectangle a  ;
  a = new rectangle( ) ;
  a.length = 5 ;
  a.breadth = 7 ;
  System.out.println( a.length ) ;
  System.out.println( a.breadth ) ;
  a.area ( ) ;
  a.perimeter ( ) ;
 }
}
```

Output :

5

7

Area is 35

Perimeter is 24

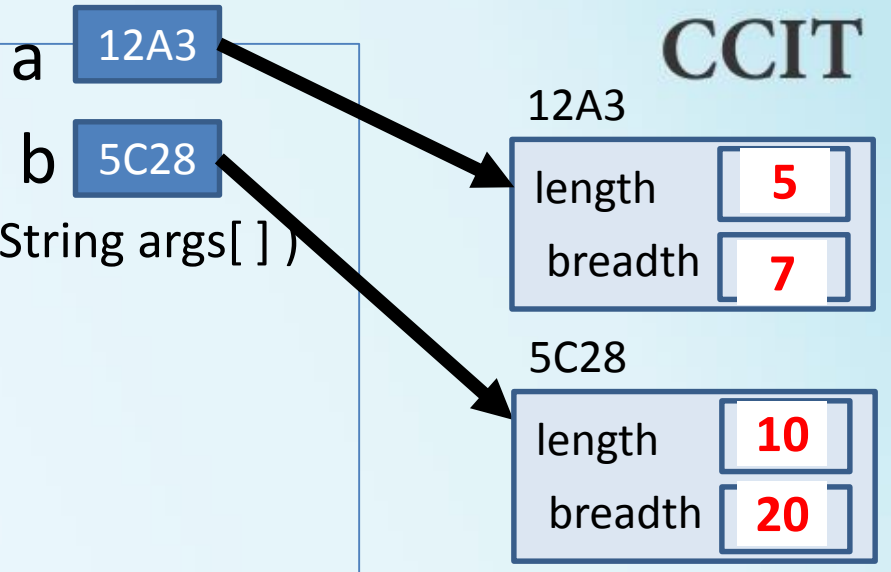
//program to create 2 rectangles a and b of size 5 x 7 and 10 x 20 . display area , perimeter of a and area of b

```
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);
        }
}
```

5 X 7

10 X 20

```
class demo
{
public static void main( String args[ ] )
  {
  rectangle a , b ;
  a = new rectangle( ) ;
  b = new rectangle( ) ;
  a.length = 5 ;
  a.breadth = 7 ;
  b.length = 10 ;
  b.breadth = 20 ;
  a.area ( ) ;
  a.perimeter ( ) ;
  b.area ( ) ;
 }
}
```



Note :
Direct access of data members must be avoided like this example. Data members must be kept private.

Output :

Area is 35
Perimeter is 24
Area is 200

Todays Topic End