Todays Topic

- Member Access Operator
- Member Access Specifiers
  - private
  - public
  - protected
  - default

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;
    }
```

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(.)

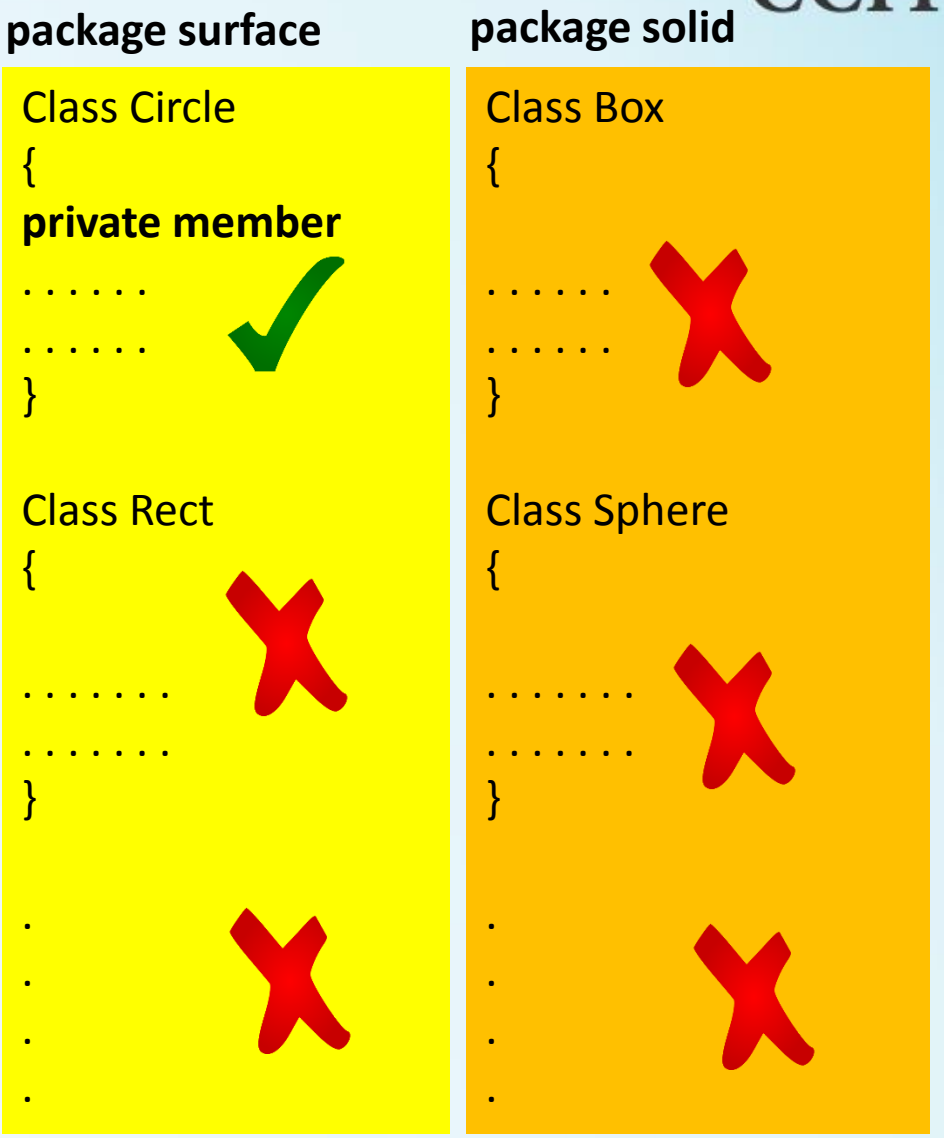
- Members of an object can be access by using member access operator
- Syntax: **ObjectName.MemberName**
- **For ex :**
- suppose **a** is an object of **class rectangle**.
  - **length** and **breadth** are data members.
  - **area** and **perimeter** are member functions**.**
- Then members of object **a** can be accessed by using member access operator.
  - **a.length**
  - **a.breadth**
  - **a.area( )**
  - **a.perimeter( )**

Note:

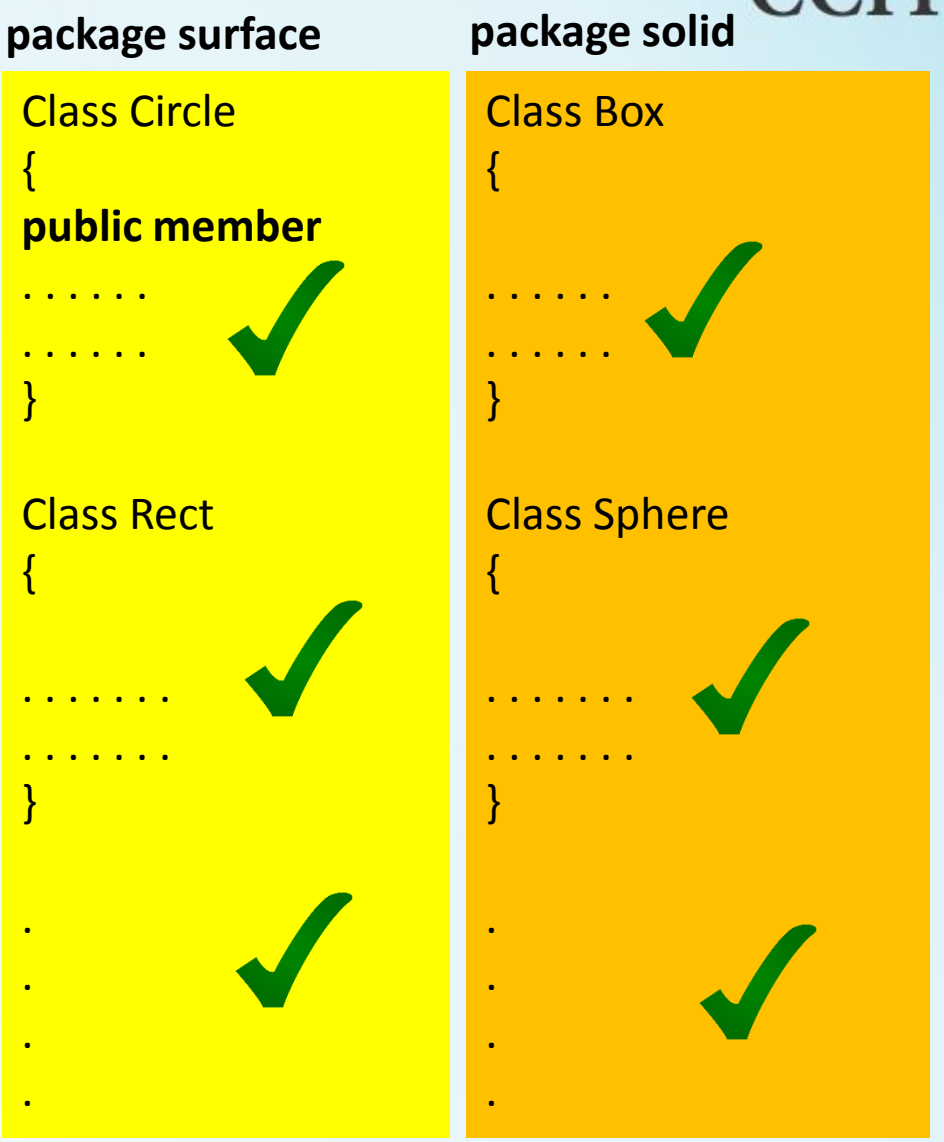
- **Basic principle of Object Oriented Programming says that data must be hidden from user of the object.**
- **This can be achieved by declaring data members as private.**

- **Access Specifiers**
- It indicates accessibility of a member of a class.
- It can be
    1. **private**
    2. **public**
    3. **(default)**
    4. **protected**

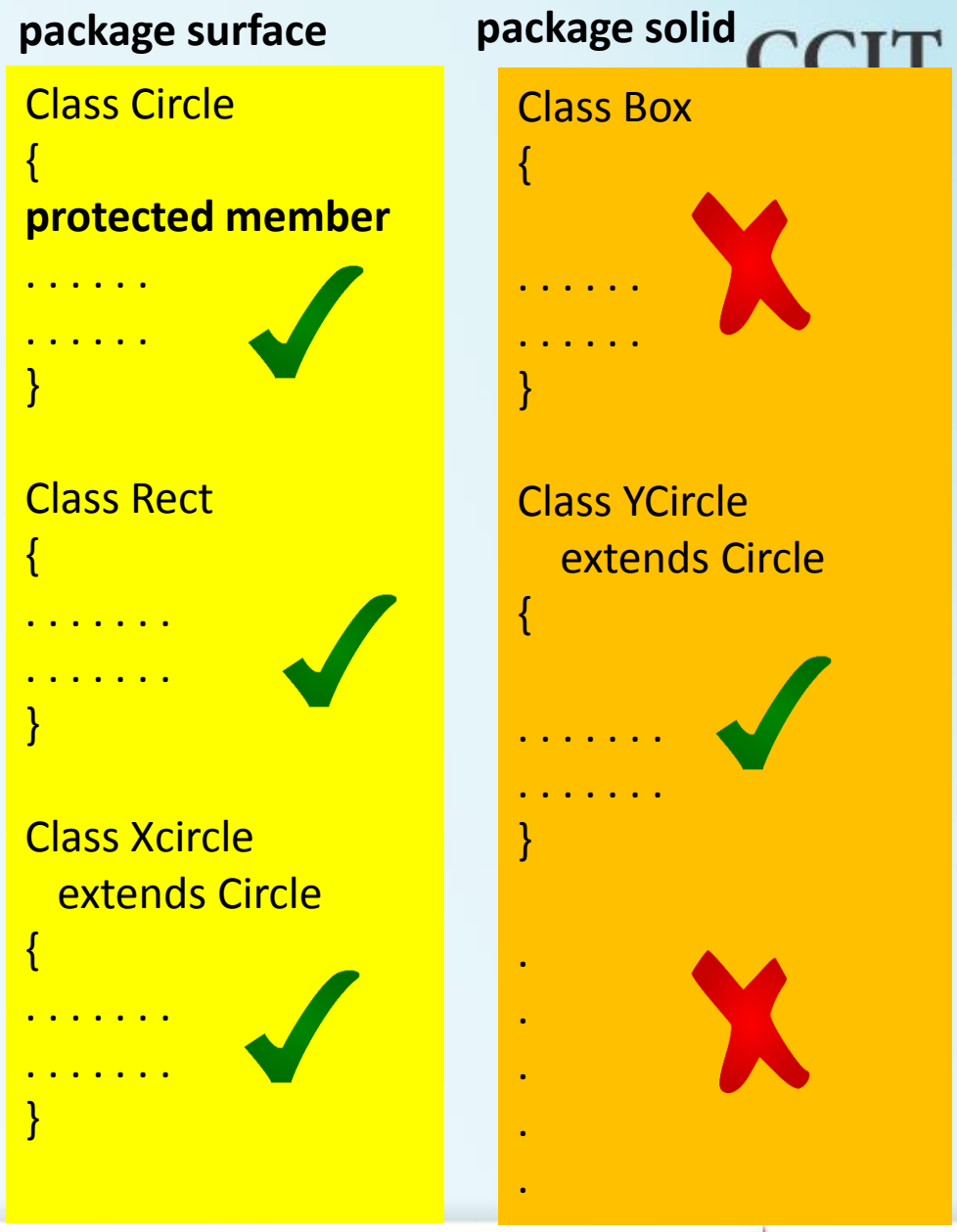
- **private members**
  - **Such members are strictly controlled, which means they cannot be accesses by anywhere outside the enclosing class.**
  - Generally data members are kept private because we want to hide internal details of object from the user of object.



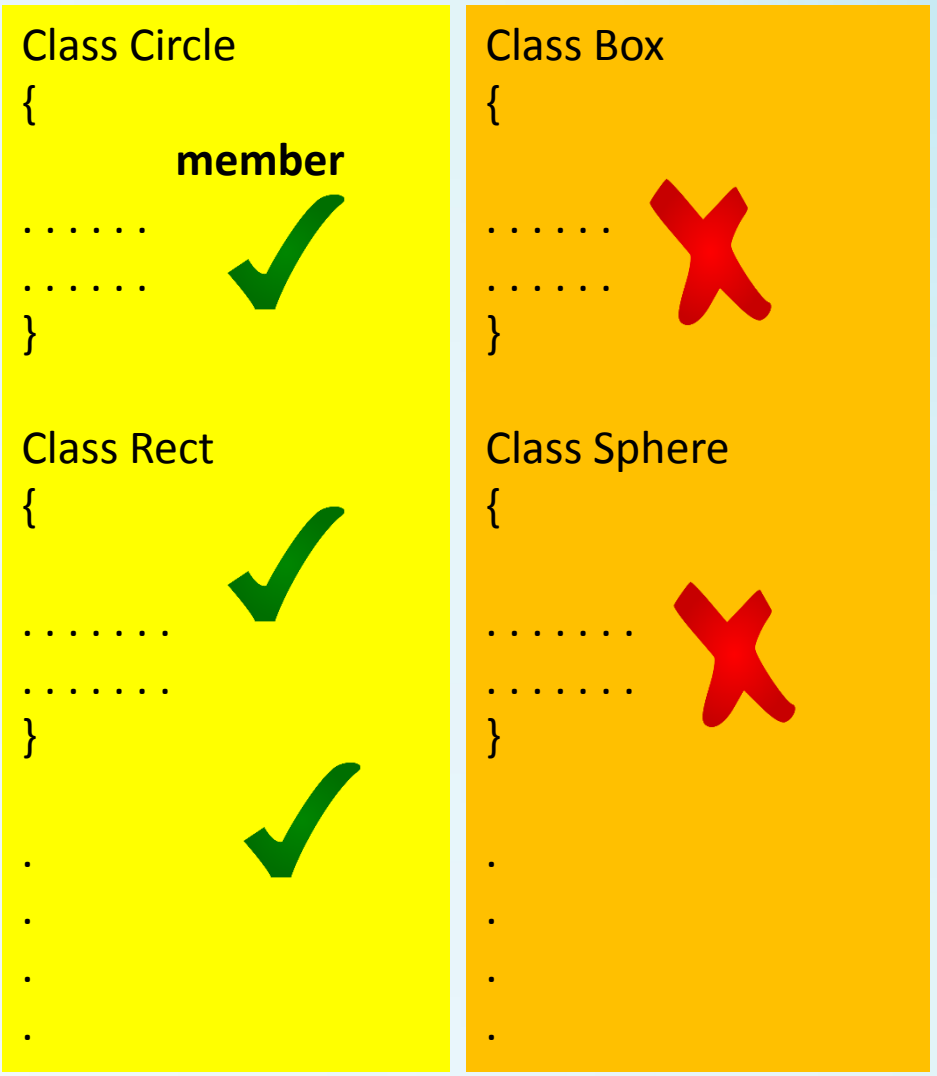
- **public members**
  - **Such members are visible to any class in the Java program, whether these classes are in the same package or in another package.**
  - Generally member functions are kept public because user wants to perform operations on object by calling its member functions.



- **protected members**
  - **Such members can be accessed by subclasses in same package as well as other packages and all classes of same package.**
  - They are used with inheritance.



- **default members**
  - **Any member that has no declared access modifier is accessible only by classes in the same package.**



Setter methods

- Generally data members are kept private.
- So we can not access such private data members outside the object.
- setter functions are used to set values for private data members of object.
- Generally their name starts with the word **set.**

Syntax:

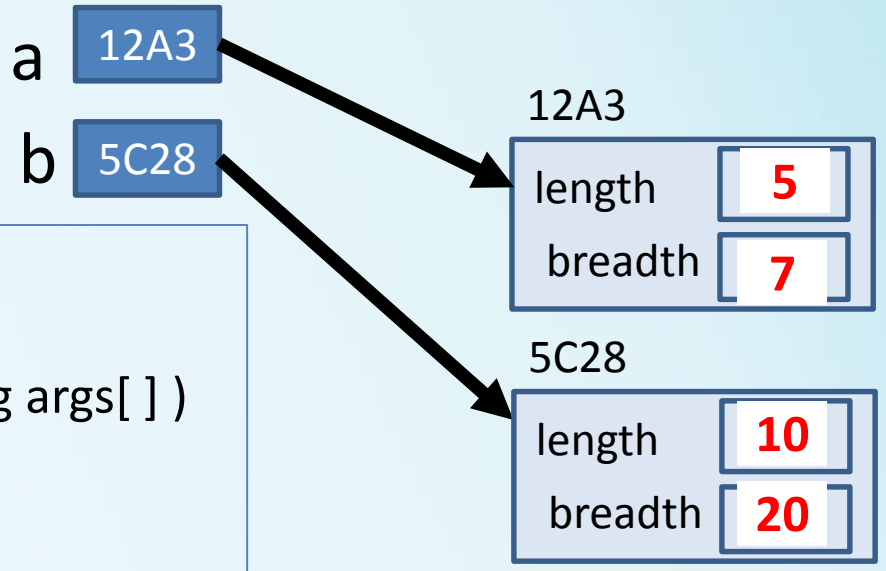
```
public void setPropertyName ( datatype arg1 , , . )
    {
    data member initialization statements
    . . . . . .
    . . . . . .
    }
```

//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
{
 private  int length , breadth;
 public  void area()
   {
    int a = length*breadth;
    System.out.println("Area is "+a)
   }
 public  void perimeter()
   {
    int p = 2*( length+breadth );
     System.out.println("perimerter is"+p);
    }
 public  void setdimension(int m,int n)
   {
    length=m;
    breadth=n;
    }
}
```

5 X 7

10 X 20




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

Output :

Area is 35

Perimeter is 24

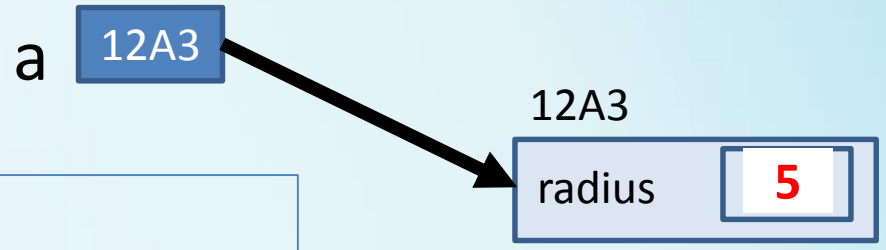
Area is 200

- Design a class Circle containing
  - data member
    - radius
  - member functions
    - area( )
    - circumference( )
    - setRadius(n)

//program to create a circle type of object of radius 5. display its area , circumference

```
class Circle
{
 private  int radius ;
 public  void area( )
  {
   double a = 3.14 * radius  * radius ;
   System.out.println("Area is "+a)
  }
 public  void circumference( )
  {
   double c = 2 * 3.14 * radius ;
    System.out.println("circumference is"+c);
   }
 public  void setRadius( int n )
  {
   radius = n ;
   }
}
```

CCIT




```
class demo
{
public static void main( String args[ ] )
  {
  Circle a ;
  a = new Circle( ) ;
  a.setRadius( 5 ) ;
  a.area( ) ;
  a.circumference( ) ;
  }
}
```

Output :

Area is 78.5

Circumference is 31.4

[AS] [M] datatype memberName [=value] ;

[AS] [M] returntype fName(datatype arg1, , .)

{

Statements…..

……

return value;

}