

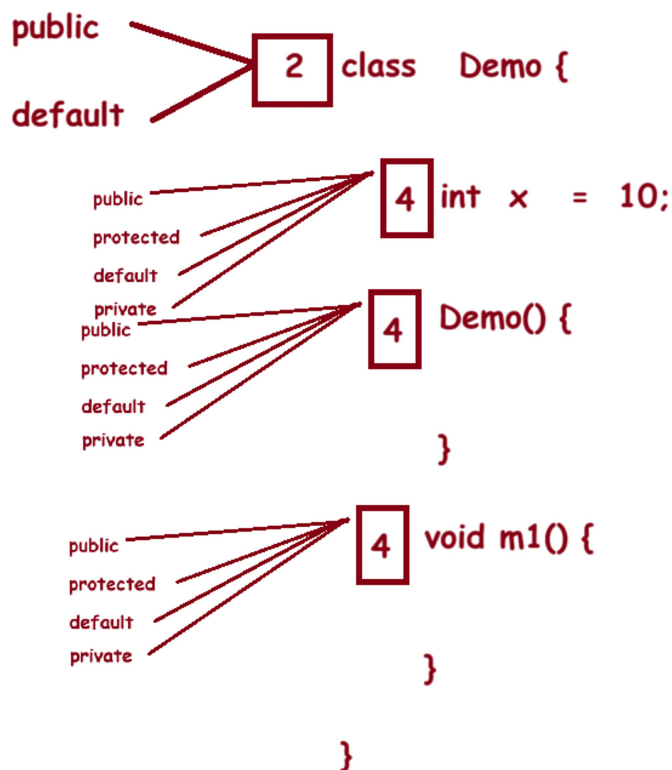
Encapsulation

What is encapsulation ?

- Encapsulation is one of the major pillar of oop's concept
- By using encapsulation we can wrap our variables and method that operates on data into a single unit
- Every java class is the example of encapsulation
- Setter and Getter (POJO) class is the perfect example encapsulation
- i.e every member in POJO class is a private data member so it will expose there own property outside the class for that we will create own method to control that property

Access modifiers

Where we can declare access modifiers



Scope of access modifiers

public :

- > public data members can be access any where
- > within a class, outside class and out side package

A.java	B.java	C.java
1 package com.adhyan;	1 package com.adhyan;	1 package com.cjc;
2	2	2
3 public class A {	3 public class B {	3 import com.adhyan.A;
4	4	4
5 public int x = 10;	5 public void m2() {	5 public class C {
6	6	6
7 public void m1() {	7 A a = new A();	7 public void m3() {
8	8 System.out.println(a.x);	8
9 System.out.println(x);	9 }	9 A a = new A();
10 }	10 }	10 System.out.println(a.x);
11 }	11	11 }
12 }		12 }
13 }		13 }
		14 }

default :

- > default data member can be access within same package
- > within class , outside class but in same package

A.java	B.java	C.java
1 package com.adhyan;	1 package com.adhyan;	1 package com.cjc;
2	2	2
3 public class A {	3 public class B {	3 import com.adhyan.A;
4	4	4
5 int x = 10;	5 public void m2() {	5 public class C {
6	6	6
7 public void m1() {	7 A a = new A();	7 public void m3() {
8	8 System.out.println(a.x);	8
9 System.out.println(x);	9 }	9 A a = new A();
10 }	10 }	10 // System.out.println(a.x); error
11 }	11	11 }
12 }		12 }
13 }		13 }
		14 }

private :

- > private data members can be access within class only

A.java	B.java	C.java
1 package com.adhyan;	1 package com.adhyan;	1 package com.cjc;
2	2	2
3 public class A {	3 public class B {	3 import com.adhyan.A;
4	4	4
5 private int x = 10;	5 public void m2() {	5 public class C {
6	6	6
7 public void m1() {	7 A a = new A();	7 public void m3() {
8	8 //System.out.println(a.x); error	8
9 System.out.println(x);	9 }	9 A a = new A();
10 }	10 }	10 // System.out.println(a.x); // error
11 }	11	11 }
12 }		12 }
13 }		13 }
		14 }

protected :

- > protected data member can be access within same package only
- > within class , outside class but in same package
- > if we want to access protected outside package then we can access in its sub class
- > we can directly in sub class or sub class object

```
A.java
1 package com.adhyan;
2
3 public class A {
4
5     protected int x = 10;
6
7     public void m1() {
8         System.out.println(x);
9     }
10 }
11
12 }
13

B.java
1 package com.adhyan;
2
3 public class B {
4
5     public void m2() {
6
7         A a = new A();
8         System.out.println(a.x);
9     }
10 }
11

C.java
1 package com.cjc;
2
3 import com.adhyan.A;
4
5 public class C extends A{
6
7     public void m3() {
8
9         A a = new A();
10        // System.out.println(a.x); error
11
12        // directly we can access
13        System.out.println(x);
14
15        // by child class object we can access
16        C c = new C();
17        System.out.println(c.x);
18    }
19 }
20 }
```