# **OOPS**

# Class Level modifiers

TopLevel class modifiers

public abstract <default> final strictfp

Innerclass class modifiers

+ private protected static

## public

|  |  |
| --- | --- |
| package pack1;  public class Student { | package pack2;  import pack1.Student;  public class Test {      public static void main(String[] args) {          Student s1= new Student(); |

If public we can access from anywhere we can access

|  |  |
| --- | --- |
| package pack1;  class Student {  public void m1() {  sop(“stud pub method”);  }  } | package pack2;  import pack1.Student;  public class Test {      public static void main(String[] args) {          Student s1= new Student();  s1.m1(); |

Public m1() can be accessed from outside package. But that class is not visible

First check **class visibility** 🡺 then **member visibility**

|  |  |
| --- | --- |
| package pack1;  public class A {  void m1() {  sop(“stud pub method”);  }  } | package pack2;  import pack1.A;  public class B {      public static void main(String[] args) {          A a= new A();  a.m1(); |

CE: m1() is not public in A; cannot be accessed from outside package

class is visible 🡺 but member(variable/ method) is not visible

## default

|  |  |
| --- | --- |
| package pack1;  class Student { | package pack2;  import pack1.Student;  public class Test {      public static void main(String[] args) {          Student s1= new Student(); |

CE: Student is not public in pack1; cannot be accessed from outside package

|  |  |
| --- | --- |
| package pack1;  class A {  public void m1() {  sop(“stud private method”);  }    void m2() {  sop(“stud private method”);  }    private void m3() {  sop(“stud private method”);  }  } | package pack1;  public class B {      public static void main(String[] args) {          A a= new A();  a.m1(); a.m2(); a.m3();  }  } |
| package pack2;  import pack1.A;  public class C {      public static void main(String[] args) {          A a= new A();  a.m1(); a.m2(); a.m3();  }  } |

**Within same package (B)**

A is visible, public member visible, default member visible m1,m2 can be access.

But private member m3 can be access within the same class only.

**From outside package (C)**

A is not visible.

Recommended modifier for **variable**(like data) is **private**.

Recommended modifier for **method**(like service) is **public**.

## abstract

|  |  |
| --- | --- |
| abstract class Vehicle {  public abstract int getWheels();  // cant do implementation  // till know the vehicle type | abstract class Loran {  public abstract double getInterest();  // cant do implementation  // till know the Loran type |

**abstract method** : the method which only has declaration but no implementation

**abstract class** : if a class contains at least one abstract method the class compulsory should be declared as abstract

For these partially implemented classes we **can’t create object**

(Instantiation is not possible)

Adapter class: these classes contain 0 abstract methods.

even though that is dummy implementations (m1(){}), so there is no use of object creation. so those classes are declared as abstract

abstract class Vehicle{

m1(){}, m2(){}, m3(){}

public abstract int getNumOfWheels();

public abstract int getNumOfSeat();

}

Child of Vehicle should implement these abstract methods otherwise child can’t create object (Restrict child to implement these methods)

~~abstract~~ class NewCar extends Vehicle{

public abstract int getNumOfWheels(){

return 3;}

}

CE : NewCar not abstract and doesn’t override getNumOfSeat()

1, implement all abstract methods

2, make this class as abstract

|  |  |
| --- | --- |
| class Car extends NewCar{  public abstract int getNumOfSeat(){  return 3;}  } | class Bus extends Vehicle{  public abstract int getNumOfWheels(){  return 6;}  public abstract int getNumOfSeat(){  return 40;}  } |

## protected

protected = <default> + kids