1. Explain about 4 access modifiers in Java Programming Language!

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Access modifier | Same class | Same package | Sub class | Other packages |
| Public | Y | Y | Y | Y |
| Protected | Y | Y | Y | N |
| No modifier | Y | Y | N | N |
| Private | Y | N | N | N |

Public -> dapat diakses oleh semua class

Protected -> dapat diakses di class sama, package sama, dan subclassnya.

No modifier -> dapat diakses di class sama dan package sama.

Private -> hanya dapat diakses di class sama.

Access modifier ini bermanfaat dalam encapsulation (information hiding). Untuk class yang tidak diturunkan, direkomendasikan untuk menggunakan access modifier private, sedangkan untuk class yang memiliki subclass, direkomendasikan untuk menggunakan access modifier protected.

2. Explain about overloading and overriding! And give example source code!

Overloading dapat terjadi ketika suatu method nama fungsi yang sama tetapi memiliki jumlah parameter / urutan parameter / tipe data parameter yang berbeda. Overloading ini bersifat static binding / trivial polymorphism dan berjalan saat compile time.

Contoh Overloading : Method print() dan method print(String name)

**public** **class** Shape {

**public** **void** print(){

System.***out***.println("This is a shape");

}

**public** **void** print(String name){

System.***out***.printf("This is a shape with name : %s",name);

}

}

Overriding dapat terjadi ketika suatu method memiliki nama fungsi dan parameter yang sama, ini dapat terjadi ketika method dari parent class dioverride di child class. Overriding ini bersifat dynamic binding / pure polymorphism dan berjalan saat run time.

Contoh Overloading : Method getArea() di class Shape dioverride oleh getArea() di class Square.

**public** **class** Shape {

**public** **double** getArea(){

**return** 0.0;

}

}

**class** Square **extends** Shape{

**private** **int** side;

@Override

**public** **double** getArea() {

**return** side \* side;

}

}

3. Mention and explain 3 methods in ArrayList class!

1. .add() -> Berfungsi untuk menambahkan elemen ke dalam ArrayList.

2. .clear() -> Berfungsi untuk mengosongkan isi dari ArrayList.

3. .size() -> Berfungsi untuk mengetahui jumlah elemen di dalam ArrayList.