Access specifiers

11/10/22

- It specifies the access, from where data member/ function can be accessed.
- It can apply to class, method, data
- 4 specifiers - Public, Private, Protected, Default
  1. Public - can be accessed from anywhere
  2. Private - Inside a class only
  3. Default - Inside same package
  4. protected -
- Refer sir's notes for sample code for access modifiers

Types of Inheritance

1. Single inheritance
2. Multilevel inheritance (not allowed in java)
3. Heirarchial inheritance

Final keyword

'final' is a keyword & it can be used with class, data member and member function

- Final variable → once data initialised, no change can be made
- Final method → Final method can't be overriden by child class
- Final class → A final class can not be inherited

Abstract class & method

Abstract method - method without body or without defination

Ex
```
class Demo {
    abstract void myfun(); // abstract method
}
```

Abstract class

- Objects of abstract class cant generated
- An abs method can have zero or more abs method

concrete method

- If there is any abstract method in a class then that class will be considered Abstract class.

Can we make a class/method final as well as abstract ?

→ NO.

- final - It is final & can not be overridden
- Abstract - It is abstract & it must be overridden in child class
- final - class can not be inherited
- Abstract - class should be inherited, if it is to be used

↳ We can store child class object in parent class reference type

Ex:

```
cl First
cl second
cl demo
    First f;
    f = new second ();
    f.fun1();
    f.fun2();
```

obj = of second

ref = of first

← stored in second class

- Any class reference type can keep the reference of,
  - ↳ its own object
  - ↳ reference of its child class object

Take a note from sir's notes

↳

Binding

- Association (linking) between method call & method defination.
- Binding takes place either at compiler or run time
  - If binding happens at a compile time then it is static binding.
  - Binding at runtime is dynamic binding.

- All final, private, static - Binding @ compilatn
- In method overriding dynamic binding occurs

Method Hiding (static method overriding)

- static method of parent class cant be overridden by child class
  - It can be done so by redifining that static method with the same signature ie. method hiding

- compiler checks for reference values' location

- Polymorphism

  - method overloading & overriding

  - refer sir's program of different shapes

  - calling calArea & printArea method for multiple shapes

Garbage Collection

- when object is of no use, that memory need to freed

Q. which objects are eligible for garbage collect^{n}?

- The objects who reference are stored nowhere

- GC done by JVM peridiocally

- for manual intervention - System.gc() or System.getRunTime().gc();