Field Class

Agenda:
1. Intro to Field Class
2. Getting Field Class objects
3. Synthetic Class Fields
4. Getting Class Field Values - (i.e Field names and their values for a instance)

-> It corresponds to the fields of a class (both Class fields (static) and Instance fields).

1. Obtaining Field Objects of a Class:

Syntax: (a) Class.getDeclaredFields() :- gets all declared fields in a class. Will return Field[]. Gives all (access Mod independent) fields of class but excludes the Inherited fields.

(b) Class.getFields() :- Gets all Public fields of a class including the inherited fields from super class.

(c) Class.getDeclaredField(String fieldName) :- Gets Field object with that name. Rules of inheritance etc are same as getDeclaredFields().

(d) Class.getField(String fieldName) :- Public field with Given name & also can look inherited field.

(2)

(III) **Synthetic Fields:** Java Compiler generates artificial fields for internal usage. These fields are invisible until we use Reflection to discover it. These fields are compiler specific & we don't touch/modify them.

To find and Check if field is Synthetic : `Field.isSynthetic();` // bool returntype

Ex: (a) Non Static Inner Class have a Synthetic Field Corresponding to the Parent Outer Class.

(b) Enums in Java also have Synthetic fields. ($value).

(IV) Getting Field Class Values for an Instance:

Syntax: `Field.get(Instance)` :- Gives the value of 'Field' on the 'instance'. If Field is Static, instance can be null too, giving correct output.

```
XYZ obj = new XYZ();
```

Ex: `Field x = XYZ.class.getDeclaredField("x");`

`x.get(obj);` // Gives the 'x' variable value in 'obj' instance.