OOP - II

Encapsulation

- Constructors
  - types of construction
  - access modifiers

Inheritance

- types of inheritance
- Diamond problem

- Diamond problem
- JAVA

Small Talk

Encapsulation

① a single unit → class / object

② Info. hiding

A a = new A()

Constructor

BankAccount {

}

Syntax

function

access ClassName (parameters)

```
class Student {
    String name;
    Integer age;
```

String batchNo;

y

Type → Default constructor

Compiler → a zero-argument constructor

public Student() {

}

Student a = new Student();

{
a.name = "EK"

a.email = ekadrona.in
}

a.name = Trump;

① unsecure "_A_"

② name → .trim

no entry point for custom logic

Getters & setters methods

- getName()
- setName() - .trim()

Student → psp - 0 - 100

- 110

- Main → entry point

Se

→ fail object creation

Constructors

A()

a.b = 1 → public

↳ setters

$ a

· A()

↳ a . a = 1

a . b = 2

↓

public

insecure

↳ setters secure

→ secure

→ validation ~~sanitisation~~

↳ allows to fail object creation

A() - zero
- nullary

Parametrised
- 1 ctor + arguments

class Student {
- int age
- string name
}

Student (age, name)

{ this.age = age || 100;
  this.name = name.trim()
  throw exception()
}

ction

1. [Compiler creates an instance
2. assign values to fields
3. [Compiler returns the instance

```
Student s1 = new Student("EK", 1);
Student s2 = new Student(s1);
```

s1.name = "Do"

s2.name => "EK"

s1.age = 20

String name

name.append()

s2.name

= 1001

access modifier

specifier

modifies the accesibility of a class method field ctor

modifies — non-access
- → final
- → static
- → synchronized

access
- → public
- → private
- → protected
- default

→ Default

↑
(B)

	class	package	subclass	global
public	Y	Y	Y	Y
private	Y	X	X	X
default (package protected)	Y	Y	X	X
protected	Y	Y	Y	X

Student

S.age

Inheritance

Alto

Ritz

Maruti800

Student
Instructor
Mentor
name
email
address
phone
duplication

User
name
email
Parent
super
base
Student
batch
psp
Instructor
ipad
job
Mentor

reusability —

- extends

class Student
extends
User {

}

Students

S. changeEmail()

① Single –

② Multilevel

③ Hierarchical

Dog — Vodafone

Java does not support multiple inheritance

Diamond problem