by Kunal Sir

this & super keyword with Inheritance

- Predict the Output Of the following question and write on notebook and after that type this code on laptop and check you predicted output and executed output same or wrong (verify your answer with laptop answer).

Problem 1 — Vehicle / Engine

```java
class Engine {
    public Engine() {
        System.out.println("Engine: default constructor");
    }
}

class Car extends Engine {
    public Car() {
        System.out.println("Car: default constructor");
    }
}

public class P1 {
    public static void main(String[] args) {
        Car myCar = new Car();
    }
}
```

Output:

```
Engine: default constructor
Car: default constructor
```

Problem 2 — User / AdminUser

```java
class User {
    public User(String role) {
        System.out.println("User: role = " + role);
    }
}

class AdminUser extends User {
    public AdminUser() {
        super("Admin");
        System.out.println("AdminUser: default constructor");
    }
}

public class P2 {
    public static void main(String[] args) {
        AdminUser admin = new AdminUser();
    }
}
```

Output :

```
User: role = Admin
AdminUser: default constructor
```

Problem 3 — Account / SavingsAccount

```
class Account {
    public Account() {
        System.out.println("Account: default");
    }
}

class SavingsAccount extends Account {
    public SavingsAccount() {
        this(1000);
        System.out.println("SavingsAccount: default");
    }

    public SavingsAccount(int balance) {
        System.out.println("SavingsAccount: balance = " + balance);
    }
}

public class P3 {
    public static void main(String[] args) {
        Account acc = new SavingsAccount();
    }
}
```

Output :

```
Account: default
SavingsAccount: balance = 1000
SavingsAccount: default
```

Problem 4 — Product / Electronics (copy constructor)

```
class Product {
    int id;

    public Product() {
        this.id = 0;
        System.out.println("Product: default id=0");
    }

    public Product(int id) {
        this.id = id;
        System.out.println("Product: id=" + id);
    }

    public Product(Product other) {
        this.id = other.id;
        System.out.println("Product: copied id=" + id);
    }
}

class Electronics extends Product {
    public Electronics() {
        super(101);
        System.out.println("Electronics: default");
    }

    public Electronics(Electronics other) {
        super(other);
        System.out.println("Electronics: copied");
    }
}

public class P4 {
    public static void main(String[] args) {
        Electronics e1 = new Electronics();
        Electronics e2 = new Electronics(e1);
    }
}
```

Output:

```
Product: id=101
Electronics: default
Product: copied id=101
Electronics: copied
```

Problem 5 — Document / SignedDocument

```
class Document {
    public Document() {
        System.out.println("Document: created");
        printStatus();
    }

    public void printStatus() {
        System.out.println("Document: status = draft");
    }
}

class SignedDocument extends Document {
    String signer = "Unknown";

    public SignedDocument() {
        System.out.println("SignedDocument: signer = " + signer);
    }

    public void printStatus() {
        System.out.println("SignedDocument: signed by " + signer);
    }
}

public class P5 {
    public static void main(String[] args) {
        Document doc = new SignedDocument();
    }
}
```

Output :

```
Document: created
SignedDocument: signed by null
SignedDocument: signer = Unknown
```

Problem 6 — Organization / Department / Team

```
class Organization {
    public Organization() {
        System.out.println("Organization: default");
    }

    public Organization(String name) {
        System.out.println("Organization: name = " + name);
    }
}

class Department extends Organization {
    public Department() {
        this(42);
        System.out.println("Department: default");
    }

    public Department(int code) {
        super("Dept-" + code);
        System.out.println("Department: code = " + code);
    }
}

class Team extends Department {
    public Team() {
        super();
        System.out.println("Team: default");
    }
}

public class P6 {
    public static void main(String[] args) {
        Team t = new Team();
    }
}
```

Output :

```
Organization: name = Dept-42
Department: code = 42
Department: default
Team: default
```

Problem 7 — Person / Employee (field shadowing)

```
class Person {
    int age = 30;

    Person() {
        System.out.println("Person.age = " + age);
    }
}

class Employee extends Person {
    int age = 25;

    Employee() {
        System.out.println("Employee.this.age = " + this.age);
        System.out.println("Employee.super.age = " + super.age);
    }
}

public class P7 {
    public static void main(String[] args) {
        Person p = new Employee();
    }
}
```

Output :

```
Person.age = 30
Employee.this.age = 25
Employee.super.age = 30
```

Problem 8 — Interface / Service / WebService

```java
interface Service {
    void info();
}

class BaseService {
    String name;

    public BaseService() {
        this.name = "BaseService";
        System.out.println("BaseService: default");
    }

    public BaseService(String name) {
        this.name = name;
        System.out.println("BaseService: name = " + name);
    }

    public BaseService(BaseService other) {
        this.name = other.name;
        System.out.println("BaseService: copied name = " + name);
    }
}

class WebService extends BaseService implements Service {
    public WebService() {
        super("WebService");
        System.out.println("WebService: default");
    }

    public void info() {
        System.out.println("WebService.info name = " + name);
    }
}

public class P8 {
    public static void main(String[] args) {
        Service s = new WebService();
        ((WebService) s).info();

        WebService ws = new WebService();
        BaseService copy = new BaseService(ws);
    }
}
```

```
Output :

BaseService: name = WebService
WebService: default
WebService.info name = WebService
BaseService: name = WebService
WebService: default
BaseService: copied name = WebService
```

Problem 9 — Retail / Cart

```java
class Item {
    public Item() {
        System.out.println("Item: default");
    }
}

class CartItem extends Item {
    public CartItem() {
        this(1);
        System.out.println("CartItem: default");
    }

    public CartItem(int qty) {
        System.out.println("CartItem: qty = " + qty);
    }
}

public class P9 {
    public static void main(String[] args) {
        CartItem c = new CartItem();
    }
}
```

```
Output :

Item: default
CartItem: qty = 1
CartItem: default
```

Problem 10 — Company / Manager / Director

```java
class Company {
    protected Company() {
        System.out.println("Company: default");
    }

    public Company(String name) {
        System.out.println("Company: name = " + name);
    }
}

class Manager extends Company {
    public Manager() {
        super("Operations");
        System.out.println("Manager: default");
    }
}

class Director extends Manager {
    public Director() {
        System.out.println("Director: default");
    }
}

public class P10 {
    public static void main(String[] args) {
        Company c = new Director();
    }
}
```

Output :

```
Company: name = Operations
Manager: default
Director: default
```

Problem 11 — DataContainer / Buffer

```
class DataContainer {
    int[] buffer;

    public DataContainer(int[] buffer) {
        this.buffer = buffer;
        System.out.println("DataContainer: created");
    }

    public DataContainer(DataContainer other) {
        this.buffer = other.buffer;
        System.out.println("DataContainer: shallow copied");
    }
}

class BufferHolder extends DataContainer {
    public BufferHolder(int[] buffer) {
        super(buffer);
        System.out.println("BufferHolder: created");
    }

    public BufferHolder(BufferHolder other) {
        super(other);
        System.out.println("BufferHolder: copied");
    }
}

public class P11 {
    public static void main(String[] args) {
        int[] arr = {1, 2, 3};
        BufferHolder h1 = new BufferHolder(arr);
        BufferHolder h2 = new BufferHolder(h1);
        h2.buffer[0] = 99;
        System.out.println("h1.buffer[0] = " + h1.buffer[0]);
    }
}
```

Output :

```
DataContainer: created
BufferHolder: created
DataContainer: shallow copied
BufferHolder: copied
h1.buffer[0] = 99
```

Problem 12 — Course / Module / Lesson

```java
class Course {
    public Course() {
        System.out.println("Course: default");
    }

    public Course(int id) {
        System.out.println("Course: id = " + id);
    }

    public void details() {
        System.out.println("Course.details");
    }
}

class Module extends Course {
    public Module() {
        this(5);
        System.out.println("Module: default");
    }

    public Module(int id) {
        super(id);
        System.out.println("Module: id = " + id);
    }

    public void details() {
        System.out.println("Module.details");
    }
}

class Lesson extends Module {
    public Lesson() {
        super();
        System.out.println("Lesson: default");
    }

    public void details() {
        System.out.println("Lesson.details");
    }
}

public class P12 {
    public static void main(String[] args) {
        Course c = new Lesson();
        c.details();
    }
}
```

```
Output :

Course: id = 5
Module: id = 5
Module: default
Lesson: default
Lesson.details
```

Problem 13 — Pricing / Discount

```
class Pricing {
    public Pricing(int amount) {
        System.out.println("Pricing: amount = " + amount);
    }
}

class Discount extends Pricing {
    public Discount() {
        this(2 + 3);
        System.out.println("Discount: default");
    }

    public Discount(int percent) {
        super(percent * 2);
        System.out.println("Discount: percent = " + percent);
    }
}

public class P13 {
    public static void main(String[] args) {
        Discount d = new Discount();
    }
}
```

```
Output :

Pricing: amount = 10
Discount: percent = 5
Discount: default
```

Problem 14 — Config / Static init / Final field

```
class Config {
    static {
        System.out.println("Config: static init");
    }

    final int version;

    public Config() {
        version = 1;
        System.out.println("Config: default version=" + version);
    }

    public Config(int v) {
        version = v;
        System.out.println("Config: version=" + version);
    }
}

class SystemConfig extends Config {
    static {
        System.out.println("SystemConfig: static init");
    }

    public SystemConfig() {
        super(10);
        System.out.println("SystemConfig: default");
    }
}

public class P14 {
    public static void main(String[] args) {
        System.out.println("Start main");
        SystemConfig sc = new SystemConfig();
    }
}
```

```
Output :

Start main
Config: static init
SystemConfig: static init
Config: version=10
SystemConfig: default
```

Problem 15 — BaseEntity / UserAccount / PremiumUser

```java
class BaseEntity {
    String id;

    public BaseEntity() {
        this.id = "base";
        System.out.println("BaseEntity: default id=" + id);
    }

    public BaseEntity(String id) {
        this.id = id;
        System.out.println("BaseEntity: id=" + id);
    }

    public BaseEntity(BaseEntity other) {
        this.id = other.id;
        System.out.println("BaseEntity: copied id=" + id);
    }

    public void greet() {
        System.out.println("Hello from " + id);
    }
}

class UserAccount extends BaseEntity {
    public UserAccount() {
        super("user");
        System.out.println("UserAccount: default");
    }

    public UserAccount(UserAccount other) {
        super(other);
        System.out.println("UserAccount: copied");
    }

    public void greet() {
        System.out.println("UserAccount greets from " + id);
    }
}

class PremiumUser extends UserAccount {
    public PremiumUser() {
        System.out.println("PremiumUser: default");
    }

    public PremiumUser(PremiumUser other) {
        super(other);
        System.out.println("PremiumUser: copied");
    }

    public void greet() {
        System.out.println("PremiumUser greets from " + id);
    }
}

public class P15 {
    public static void main(String[] args) {
        BaseEntity b = new PremiumUser();
        UserAccount u = new PremiumUser();
        PremiumUser p1 = new PremiumUser();
        PremiumUser p2 = new PremiumUser(p1);
        BaseEntity br = p2;

        b.greet();
        u.greet();
        p1.greet();
        p2.greet();
        br.greet();
    }
}
```

Output :

```
BaseEntity: id=user
UserAccount: default
PremiumUser: default
BaseEntity: id=user
UserAccount: default
PremiumUser: default
BaseEntity: id=user
UserAccount: default
PremiumUser: default
BaseEntity: copied id=user
UserAccount: copied
PremiumUser: copied
PremiumUser greets from user
PremiumUser greets from user
PremiumUser greets from user
PremiumUser greets from user
PremiumUser greets from user
```