

Exercise 1: Marks 10

Given the piece of code make a class diagram and calculate the OO metrics including

1. Size metrics
2. Coupling metrics
3. Cohesion metrics
4. Inheritance metrics

```
class Animal {
    protected String species;
    protected int age;
    protected boolean isPet;
    public Animal(String species, int age, boolean isPet) {
        this.species = species;
        this.age = age;
        this.isPet = isPet;
    }
    public void makeSound() {
        System.out.println( " Animal sound " );
    }
    public void printInfo() {
        System.out.println( " Species: " + species + " ,
Age: " + age +
        &
        " , Is a Pet: " + isPet );
    }
}
class Dog extends Animal {

    private String breed;
    private String color;
    public Dog(String breed, String color, int age, boolean isPet) {
        super( " Dog " , age, isPet );
        this.breed = breed;
        this.color = color;
    }
    @Override
    public void makeSound() {
        System.out.println( " Woof! " );
    }
    public void printBreed() {
        System.out.println( " Breed: " + breed );
    }
    public void printColor() {
```

```

        System.out.println( & quot; Color: & quot; + color);
    }
}
class Cat extends Animal {
    private String furType;
    private boolean isOutdoor;
    public Cat(String furType, boolean isOutdoor, int age, boolean isPet) {
        super( & quot; Cat & quot;; age, isPet);
        this.furType = furType;
        this.isOutdoor = isOutdoor;
    }
    @Override
    public void makeSound() {
        System.out.println( & quot; Meow! & quot;);
    }
    public void printFurType() {

        System.out.println( & quot; Fur Type: & quot; + furType);
    }
    public void printOutdoorStatus() {
        System.out.println( & quot; Outdoor: & quot; + isOutdoor);
    }
}
class Bird extends Animal {
    private boolean canFly;
    public Bird(boolean canFly, int age, boolean isPet) {
        super( & quot; Bird & quot;; age, isPet);
        this.canFly = canFly;
    }
    @Override
    public void makeSound() {
        System.out.println( & quot; Chirp! & quot;);
    }
    public void printFlightCapability() {
        System.out.println( & quot; Can Fly: & quot; + canFly);
    }
}
}

```

Exercise 2: Marks 10

Given the piece of code make a class diagram and calculate the OO metrics including

1. Size metrics
2. Coupling metrics
3. Cohesion metrics

4. Inheritance metrics

```
class Engine {
    private String type;
    private int horsepower;

    public Engine(String type, int horsepower) {
        this.type = type;
        this.horsepower = horsepower;
    }
    public void start() {
        System.out.println( & quot; Engine started & quot;);
    }
    public void printInfo() {
        System.out.println( & quot; Engine Type: & quot; + type + & quot;;
Horsepower: & quot; +
        horsepower);
    }
}
class Car {
    private String brand;
    private Engine engine;
    private boolean isElectric;
    public Car(String brand, String engineType, int horsepower, boolean
isElectric) {
        this.brand = brand;
        this.engine = new Engine(engineType, horsepower);
        this.isElectric = isElectric;
    }
    public void start() {
        System.out.println( & quot; Car started & quot;);
        engine.start();
    }
    public void printInfo() {
        System.out.println( & quot; Car Brand: & quot; + brand + & quot;;
Is Electric: & quot; +
        isElectric);
        engine.printInfo();
    }
}
class Motorcycle {
    private String brand;
    private Engine engine;
```

```
private String bikeType;

public Motorcycle(String brand, String engineType, int horsepower,
    String bikeType) {
    this.brand = brand;
    this.engine = new Engine(engineType, horsepower);
    this.bikeType = bikeType;
}
public void start() {
    System.out.println( " Motorcycle started " );
    engine.start();
}
public void printInfo() {
    System.out.println( " Motorcycle Brand: " + brand + "
quot;, Bike Type: " +
        bikeType);
    engine.printInfo();
}
}
```