**OOP concepts in java**

**Adeel Mazhar**

**Class and Object:**

class Car {

String brand;

int year;

void displayInfo() {

System.out.println("Brand: " + brand + ", Year: " + year);

}

}

public class Main {

public static void main(String[] args) {

Car myCar = new Car();

myCar.brand = "Toyota";

myCar.year = 2022;

myCar.displayInfo();

}

}

**}**

**Encapsulation:**

**class Student {**

**private String name;**

**public void setName(String newName) {**

**name = newName;**

**}**

**public String getName() {**

**return name;**

**}**

**}**

**public class Main {**

**public static void main(String[] args) {**

**Student student = new Student();**

**student.setName("John");**

**System.out.println("Student Name: " + student.getName());**

**}**

**}**

**Inheritance:**

**class Animal {**

**void eat() {**

**System.out.println("Animal is eating");**

**}**

**}**

**class Dog extends Animal {**

**void bark() {**

**System.out.println("Dog is barking");**

**}**

**}**

**public class Main {**

**public static void main(String[] args) {**

**Dog myDog = new Dog();**

**myDog.eat();**

**myDog.bark();**

**}**

**}**

**Polymorphism:**

class Shape {

void draw() {

System.out.println("Drawing a shape");

}

}

class Circle extends Shape {

@Override

void draw() {

System.out.println("Drawing a circle");

}

}

class Square extends Shape {

@Override

void draw() {

System.out.println("Drawing a square");

}

}

public class Main {

public static void main(String[] args) {

Shape circle = new Circle();

Shape square = new Square();

circle.draw();

square.draw();

}

}