**1.**

import java.time.LocalDateTime;

import java.time.format.DateTimeFormatter;

public class DatenTime {

public static void main(String[] args) {

    LocalDateTime Myformatobj = LocalDateTime.now();

    System.out.println("Before Formatting: "+Myformatobj);

    DateTimeFormatter obj2 = DateTimeFormatter.ofPattern("dd-MM-YYYY HH:MM:SS");

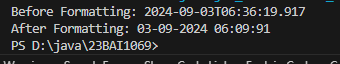
    String Formatted = Myformatobj.format(obj2);

    System.out.println("After Formatting: "+ Formatted);

}

}

**Output:**

****

**2.**

class Employee{

    float salary= 40000;

}

class Inherit1 extends Employee {

    int bonus = 10000;

    public static void main(String[] args) {

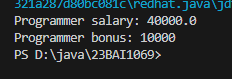
        Inherit1 p = new Inherit1();

        System.out.println(("Programmer salary: "+ p.salary + "\nProgrammer bonus: "+ p.bonus));

    }

}

**Output:**

****

**3.**

class Animal{

    void eat(){System.out.println("Eating...");

}

class Dog extends Animal{

void bark(){System.out.println("Barking...");}

}

class Puppy extends Dog{

    void weep(){System.out.println("Weeping...");

    }

}

        public class Mulinherit {

            public static void main(String[] args) {

            Puppy puppy = new Puppy();

            puppy.weep();

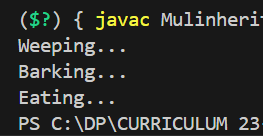
            puppy.bark();

            puppy.eat();

            }

}

**Output:**

****

**4.**

class Animal{

 void eat(){System.out.println("Eating..");}

}

class Dog extends Animal{

    void bark(){System.out.println("Barking...");}

}

class Cat extends Animal{

    void meow(){System.out.println("Meow..");}

    }

public class Inherit2 {

  public static void main(String[] args) {

    Dog d = new Dog();

    Cat c= new Cat();

    c.meow();

    c.eat();

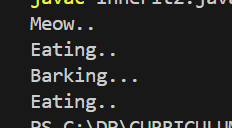
    d.bark();

    d.eat();

  }

}

**Output:**

****

**5. Exercise**

import java.util.Scanner;

public class AP {

    static int nthTerm(int f, int n, int d){

        int nterm = f+ (n-1)\*d;

        return nterm;

    }

    static int posn(int f, int k, int d){

        if((k-f)%d==0) {

            int r = (k-f)/d + 1;

            return r;}

        else return 0;

    }

    static int abval(int a, int b){

        int s = Math.abs(a-b);

        return s;

    }

    public static void main(String[] args) {

     Scanner sc = new Scanner(System.in);

      int f = sc.nextInt();

      int d = sc.nextInt();

      int n = sc.nextInt();

      int k = sc.nextInt();

      int p = sc.nextInt();

      int q = sc.nextInt();

      int ans1 = nthTerm(f,n,d);

      int ans2 = posn(f,k,d);

      int ap = nthTerm(f,p,d);

      int aq = nthTerm(f,q,d);

      int ans3 = abval(ap,aq);

      System.out.print(ans1);

      System.out.print("  " +ans2);

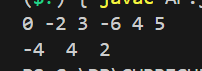
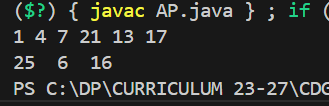
      System.out.print("  " +ans3);

      }

}

// a+ n-1\* d = (k-f)/d + 1

**Output:**

** **