//1.

import java.util.Scanner;

class Student\_Grade

{

public static void main(String[] args)

{

Scanner input = new Scanner(System.in);

System.out.println("Enter The Five Subject Marks :");

int m1 = input.nextInt();

int m2 = input.nextInt();

int m3 = input.nextInt();

int m4 = input.nextInt();

int m5 = input.nextInt();

int tot = m1+m2+m3+m4+m5;

float per = (tot/500)\*100;

System.out.println("Total :"+tot);

System.out.println("Percentage :"+per);

if(per>=90 && per<=100)

System.out.println("Grade A");

else if(per>=80 && per<=89)

System.out.println("Grade B");

else if(per>=70 && per<=79)

System.out.println("Grade C");

else if(per>=60 && per<=69)

System.out.println("Grade D");

else if(per>=40)

System.out.println("Grade E");

else

System.out.println("Grade F");

}

}

//2.

public class Person {

// Fields

private String name;

private int age;

// Constructor

public Person(String name, int age) {

this.name = name;

this.age = age;

}

// Method to display information

public void displayInfo() {

System.out.println("Name: " + name);

System.out.println("Age: " + age);

}

public static void main(String[] args) {

// Creating an instance of Person using the constructor

Person person1 = new Person("John Doe", 25);

// Displaying information using the displayInfo method

person1.displayInfo();

}

}

//3.

class Animal {

void eat() {

System.out.println("The animal is eating");

}

void sleep() {

System.out.println("The animal is sleeping");

}

}

// Child class inheriting from Animal

class Dog extends Animal {

void bark() {

System.out.println("The dog is barking");

}

}

// Another child class inheriting from Animal

class Cat extends Animal {

void meow() {

System.out.println("The cat is meowing");

}

}

// Main class to test the inheritance

public class InheritanceExample {

public static void main(String[] args) {

// Creating objects of the child classes

Dog myDog = new Dog();

Cat myCat = new Cat();

// Calling methods from the parent class

myDog.eat();

myDog.sleep();

// Calling methods from the child class

myDog.bark();

// Calling methods from another child class

myCat.eat();

myCat.sleep();

myCat.meow();

}

}

//4.

import java.util.Scanner;

public class FactorialCalculator {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter a non-negative integer: ");

int n = scanner.nextInt();

if (n < 0) {

System.out.println("Please enter a non-negative integer.");

} else {

long factorial = fact(n);

System.out.println("Factorial of " + n + " is: " + factorial);

}

scanner.close();

}

private static long fact(int n) {

if (n == 0 || n == 1) {

return 1;

} else {

return n \* fact(n - 1);

}

}

}

//5.

import java.util.Scanner;

public class DayOfWeek {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter a number (1-7): ");

int dayNumber = scanner.nextInt();

String day;

switch (dayNumber) {

case 1:

day = "Sunday";

break;

case 2:

day = "Monday";

break;

case 3:

day = "Tuesday";

break;

case 4:

day = "Wednesday";

break;

case 5:

day = "Thursday";

break;

case 6:

day = "Friday";

break;

case 7:

day = "Saturday";

break;

default:

day = "Invalid day number";

}

System.out.println("Day of the week: " + day);

}

}

//6.

class Shape {

public void draw() {

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

}

}

class Circle extends Shape {

@Override

public void draw() {

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

}

}

class Square extends Shape {

@Override

public void draw() {

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

}

}

public class PolymorphismExample {

public static void main(String[] args) {

Shape shape1 = new Circle();

Shape shape2 = new Square();

// Polymorphism in action

shape1.draw(); // Calls the draw method of Circle

shape2.draw(); // Calls the draw method of Square

}

}

//7.

import java.util.Scanner;

public class PalindromeString {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter a string: ");

String original = scanner.nextLine();

String rev = "";

for (int i =original.length()-1; i >= 0; i--) {

rev = rev + original.charAt(i);

}

if (original.equals(rev)) {

System.out.println("palindrome");

} else {

System.out.println("not a palindrome string");

}

}

}