# Level 2 Practice Programs

1. Create a program to find the factors of a number taken as user input, store the factors in an array, and display the factors. Also find the sum, sum of square of factors and product of the factors and display the results

**Hint =>**

1. Take the input for a number
2. Write a ***static*** Method to find the factors of the number and save them in an array and return the array.
3. To find factors and save to array will have two loops. The first loop to find the count and initialize the array with the count. And the second loop save the factors into the array
4. Write a method to find the sum of the factors using factors array
5. Write a method to find the product of the factors using factors array
6. Write a method to find the sum of square of the factors using ***Math.pow()*** method

import java.util.Scanner;

class Factors {

public static int[] factors(int n) {

int count = 0;

for (int i = 1; i <= n; i++) {

if (n % i == 0) {

count++;

}

}

int[] factors = new int[count];

int index = 0;

for (int i = 1; i <= n; i++) {

if (n % i == 0) {

factors[index] = i;

index++;

}

}

return factors;

}

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.println("Enter a number: ");

int num = sc.nextInt();

int[] factors = factors(num);

System.out.println("The factors of " + num + " are:");

for (int factor : factors) {

System.out.println(factor);

}

}

}

1. Write a program to find the sum of n natural numbers using recursive method and compare the result with the formulae n\*(n+1)/2 and show the result from both computations is correct.

**Hint =>**

1. Take the user input number and check whether it's a Natural number, if not exit
2. Write a Method to find the sum of n natural numbers using **recursion**
3. Write a Method to find the sum of n natural numbers using the formulae n\*(n+1)/2
4. Compare the two results and print the result

import java.util.Scanner;

public class AthleteRun {

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

int num=input.nextInt();

System.out.println(sum(num));

System.out.println(sum1(num));

}

public static int sum(int num){

if (num==0){

return 0;

}

return num+sum(num-1);

}

public static int sum1(int n){

return n\*(n+1)/2;

}

}

1. Write a program that takes a year as input and outputs the Year is a Leap Year or not

**Hint =>**

1. The LeapYear program only works for year >= 1582, corresponding to a year in the Gregorian calendar.
2. Also Leap year is divisible by 4 and not divisible by 100 or divisible by 400
3. Write a method to check for Leap Year using the conditions a and b

import java.util.Scanner;

public class AthleteRun {

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

int num=input.nextInt();

System.out.println(Leapyear(num));

}

public static boolean Leapyear(int y){

if(y>=1582){

if(y%4==0 && y%100!=0){

return y%400!=0;

}

return false;

}

return false;

}

}

1. Extend or Create a ***UnitConvertor*** utility class similar to the one shown in the notes to do the following. Please define ***static*** methods for all the UnitConvertor class methods. E.g.

***public static double convertKmToMiles(double km)* =>**

1. Method To convert kilometers to miles and return the value. Use the following code double km2miles = 0.621371;
2. Method to convert miles to kilometers and return the value. Use the following code double miles2km = 1.60934;
3. Method to convert meters to feet and return the value. Use the following code to convert double meters2feet = 3.28084;
4. Method to convert feet to meters and return the value. Use the following code to convert double feet2meters = 0.3048;

import java.util.Scanner;

public class AthleteRun {

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

int num=input.nextInt();

UnitConverter one=new UnitConverter();

System.out.println(one.KmToMiles(num));

System.out.println(one.milesToKm(num));

System.out.println(one.metersToFeet(num));

System.out.println(one.feetToMeters(num));

}

}

public class UnitConverter{

public static double KmToMiles(double km){

return km\*0.621371;

}

public static double milesToKm(double m){

return m\*1.60934;

}

public static double metersToFeet(double meter){

return meter\*3.28084;

}

public static double feetToMeters(double f){

return f\*0.3048;

}

}

1. Extend or Create a ***UnitConvertor*** utility class similar to the one shown in the notes to do the following. Please define ***static*** methods for all the UnitConvertor class methods. E.g.

***public static double convertYardsToFeet(double yards)* =>**

1. Method to convert yards to feet and return the value. Use following code to convert double yards2feet = 3;
2. Method to convert feet to yards and return the value. Use following code to convert double feet2yards = 0.333333;
3. Method to convert meters to inches and return the value. Use following code to convert double meters2inches = 39.3701;
4. Method to convert inches to meters and return the value. Use following code to convert double inches2meters = 0.0254;
5. Method to convert inches to centimeters and return the value. Use the following code double inches2cm = 2.54;

import java.util.Scanner;

public class AthleteRun {

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

int num=input.nextInt();

UnitConverter one=new UnitConverter();

System.out.println(one.yardsToFeet(num));

System.out.println(one.feetToYards(num));

System.out.println(one.metersToInches(num));

System.out.println(one.inchesToMeters(num));

System.out.println(one.inchesToCm(num));

}

}

public class UnitConverter{

public static double yardsToFeet(double y){

return y\*3;

}

public static double feetToYards(double f){

return f\*0.333333;

}

public static double metersToInches(double m){

return m\*39.3701;

}

public static double inchesToMeters(double i){

return i\*0.0254 ;

}

public static double inchesToCm(double in){

return in\*2.54;

}

}

1. Extend or Create a ***UnitConvertor*** utility class similar to the one shown in the notes to do the following. Please define ***static*** methods for all the UnitConvertor class methods. E.g.

***public static double convertFarhenheitToCelsius(double farhenheit)* =>**

1. Method to convert Fahrenheit to Celsius and return the value. Use the following code double farhenheit2celsius = (farhenheit - 32) \* 5 / 9;
2. Method to convert Celsius to Fahrenheit and return the value. Use the following code double celsius2farhenheit = (celsius \* 9 / 5) + 32;
3. Method to convert pounds to kilograms and return the value. Use the following code double pounds2kilograms = 0.453592;
4. Method to convert kilograms to pounds and return the value. Use the following code double kilograms2pounds = 2.20462;
5. Method to convert gallons to liters and return the value. Use following code to convert double gallons2liters = 3.78541;
6. Method to convert liters to gallons and return the value. Use following code to convert double liters2gallons = 0.264172;

import java.sql.SQLOutput;

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

double a = sc.nextDouble();

UnitConverter uc = new UnitConverter();

System.out.println(uc.far2celsius(a));

System.out.println(uc.celsius2far(a));

System.out.println(uc.pounds2kilo(a));

System.out.println(uc.kilo2pounds(a));

System.out.println(uc.gallon2litre(a));

System.out.println(uc.litre2gallon(a));

}

}

class UnitConverter{

public static double far2celsius(double far){

return (far-32)\*5/9;

}

public static double celsius2far(double celsius){

return (celsius\*9/5)+32;

}

public static double pounds2kilo(double pounds){

return pounds\*0.45359237;

}

public static double kilo2pounds(double kilo){

return kilo\*2.20462;

}

public static double gallon2litre(double gal) {

return gal\*3.78541;

}

public static double litre2gallon(double litre) {

return litre\*0.264172;

}

}

1. Write a program to take user input for the age of all 10 students in a class and check whether the student can vote depending on his/her age is greater or equal to 18.

**Hint =>**

1. Create a class ***public class StudentVoteChecker*** and define a method ***public boolean canStudentVote(int age)*** which takes in age as a parameter and returns true or false
2. Inside the method firstly validate the age for a negative number, if a negative return is false cannot vote. For valid age check for age is 18 or above return true; else return false;
3. In the main function define an array of 10 integer elements, loop through the array by take user input for the student's age, call canStudentVote() and display the result

import java.sql.SQLOutput;

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

int student;

for(int i=0;i<10;i++){

student = sc.nextInt();

System.out.println(StudentVoteChecker(student));

}

}

public static String StudentVoteChecker(int age){

if(age>=18){

return "Valid";

}

else{

return "Invalid";

}

}

}

1. Create a program to find the youngest friends among 3 Amar, Akbar and Anthony based on their ages and tallest among the friends based on their heights and display it

**Hint =>**

1. Take user input for age and height for the 3 friends and store it in two arrays each to store the values for age and height of the 3 friends
2. Write a Method to find the youngest of the 3 friends
3. Write a Method to find the tallest of the 3 friends

import java.util.Scanner;

public class FriendsComparison {

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

String[] names = {"Amar", "Akbar", "Anthony"};

int[] ages = new int[3];

double[] heights = new double[3];

for (int i = 0; i < 3; i++) {

System.out.print("Enter age of " + names[i] + ": ");

ages[i] = input.nextInt();

System.out.print("Enter height of " + names[i] + " (in cm): ");

heights[i] = input.nextDouble();

}

String youngest = findYoungest(names, ages);

String tallest = findTallest(names, heights);

System.out.println("Youngest friend: " + youngest);

System.out.println("Tallest friend: " + tallest);

input.close();

}

public static String findYoungest(String[] names, int[] ages) {

int minAge = ages[0];

int index = 0;

for (int i = 1; i < ages.length; i++) {

if (ages[i] < minAge) {

minAge = ages[i];

index = i;

}

}

return names[index];

}

public static String findTallest(String[] names, double[] heights) {

double maxHeight = heights[0];

int index = 0;

for (int i = 1; i < heights.length; i++) {

if (heights[i] > maxHeight) {

maxHeight = heights[i];

index = i;

}

}

return names[index];

}

}

1. Write a program to take user input for 5 numbers and check whether a number is positive or negative. Further for positive numbers check if the number is even or odd. Finally compare the first and last elements of the array and display if they are equal, greater, or less

**Hint =>**

1. Write a Method to Check whether the number is positive or negative
2. Write a Method to check whether the number is even or odd
3. Write a Method to compare two numbers and return 1 if number1 > number2 or 0 if both are equal or -1 if number1 < number2
4. In the main program, Loop through the array using the length call the method ***isPositive()*** and if positive call method ***isEven()*** and print accordingly
5. If the number is negative, print negative.
6. Finally compare the first and last element of the array by calling the method ***compare()*** and display if they are equal, greater, or less

import java.util.Scanner;

class Check {

public static String isPositive(int n) {

if (n>0){

return "Positive";

}

else{

return "Negative";

}

}

public static String evenOrOdd(int n) {

if (n%2==0){

return "Even";

}

else{

return "Odd";

}

}

public static String greaterThan(int n,int m) {

if(n>m){

return "num1 is Greater";

}

else{

return "num2 is Greater";

}

}

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

int[] arr = new int[5];

for (int i = 0; i < 5; i++) {

arr[i] = sc.nextInt();

System.out.println(isPositive(arr[i]));

System.out.println(evenOrOdd(arr[i]));

}

System.out.println(greaterThan(arr[0],arr[4]));

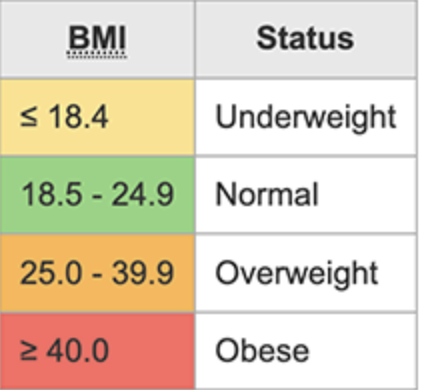
}

}

1. An organization took up the exercise to find the Body Mass Index (BMI) of all the persons in the team of 10 members. For this create a program to find the BMI and display the height, weight, BMI and status of each individual

**Hint =>**

1. Take user input in double for the weight (in kg) of the person and height (in cm) for the person and and store it in the corresponding 2D array of 10 rows and 3 columns. The First Column storing the weight, the second column storing the height in cm and the third column is the BMI
2. Create a Method to find the BMI of every person and populate the array. Use the formula BMI = weight / (height \* height). Note unit is kg/m^2. For this convert cm to meter
3. Create a Method to determine the BMI status using the logic shown in the figure below. and return the array of all the persons BMI Status.



import java.util.Scanner;

class Check {

public static String BMI(int weight, int height) {

double bmi = weight / Math.pow(((double) height / 100), 2);

if (bmi <= 18.4) {

return "Underweight";

} else if (bmi <= 24.9) {

return "Normal";

} else if (bmi <= 39.9) {

return "Overweight";

} else if (bmi > 40) {

return String.format("Obese (BMI: %.2f)", bmi);

}

return "Unable to determine BMI classification";

}

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

int[][] arr = new int[10][3];

for (int i = 0; i < 10; i++) {

arr[i][0] = sc.nextInt();

arr[i][1] = sc.nextInt();

System.out.println(BMI(arr[i][0], arr[i][1]));

}

}

}

1. Write a program Quadratic to find the roots of the equation . Use Math functions ***Math.pow()*** and ***Math.sqrt()***

**Hint =>**

1. Take a, b, and c as input values to find the roots of x.
2. The roots are computed using the following formulae

If delta is positive the find the two roots using formulae

If delta is zero then there is only one root of x

If delta is negative return empty array or nothing

1. Write a Method to find find the roots of a quadratic equation and return the roots

import java.util.Scanner;

class Check {

// Method returns an array of doubles for the roots

public static double[] roots(int a, int b, int c) {

double delta = (b \* b) - 4 \* a \* c; // Calculate delta (discriminant)

if (delta < 0) {

// No real roots exist

System.out.println("No real roots.");

return new double[0]; // Return an empty array

} else if (delta == 0) {

// One real root (delta = 0)

double root = -b / (2.0 \* a);

return new double[] {root}; // Return single root as array

} else {

// Two distinct real roots (delta > 0)

double root1 = (-b + Math.sqrt(delta)) / (2 \* a);

double root2 = (-b - Math.sqrt(delta)) / (2 \* a);

return new double[] {root1, root2}; // Return roots as array

}

}

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

// Input values for a, b, c

System.out.println("Enter values for a, b, and c:");

int a = sc.nextInt();

int b = sc.nextInt();

int c = sc.nextInt();

// Call the roots method and print results

double[] result = roots(a, b, c);

// Display the roots

if (result.length == 0) {

System.out.println("No real roots found.");

} else if (result.length == 1) {

System.out.println("Root: " + result[0]);

} else {

System.out.println("Root 1: " + result[0]);

System.out.println("Root 2: " + result[1]);

}

}

}

1. Write a program that generates five 4 digit random values and then finds their average value, and their minimum and maximum value. Use Math.random(), Math.min(), and Math.max().

**Hint =>**

1. Write a method that generates array of 4 digit random numbers given the size as a parameter as shown in the method signature

***public int[] generate4DigitRandomArray(int size)***

1. Write a method to find average, min and max value of an array

***public double[] findAverageMinMax(int[] numbers)***

import java.util.Random;

public class RandomNumbers {

public static void main(String[] args) {

int size = 5;

int[] numbers = generate4DigitRandomArray(size);

System.out.print("Generated Numbers: ");

for (int num : numbers) {

System.out.print(num + " ");

}

double[] results = findAverageMinMax(numbers);

System.out.println("\nAverage: " + results[0]);

System.out.println("Minimum: " + results[1]);

System.out.println("Maximum: " + results[2]);

}

public static int[] generate4DigitRandomArray(int size) {

int[] arr = new int[size];

Random rand = new Random();

for (int i = 0; i < size; i++) {

arr[i] = rand.nextInt(9000) + 1000; // Generates a 4-digit number (1000 - 9999)

}

return arr;

}

public static double[] findAverageMinMax(int[] numbers) {

double sum = 0;

int min = numbers[0];

int max = numbers[0];

for (int num : numbers) {

sum += num;

min = Math.min(min, num);

max = Math.max(max, num);

}

double avg = sum / numbers.length;

return new double[]{avg, min, max};

}

}