import java.util.Scanner;

import java.io.IOException;

import java.io.OutputStream;

import java.io.PrintStream;

import java.io.File;

import java.io.PrintWriter;

import java.text.DecimalFormat;

/\*\*

\*

\* This program reads a student’s name, semester letter grades, and semester hours from one file,

\* calculates the semester GPA; and writes the student’s name and semester GPA to another file

\*

\* @author Mehmet ACIKGOZ

\*

\*/

public class CalculateGPA {

/\*\*

\* This program reads a student’s name, semester letter grades, and semester hours from one file,

\* calculates the semester GPA; and writes the student’s name and semester GPA to another file

\* @param args

\* @throws IOException

\*/

public static void main(String[] args) throws IOException

{

Scanner inFile = openFile();

String[] gradeSummary = processGrades(inFile);

inFile.close();

storeGpa(gradeSummary);

System.out.println("Data processing complete.");

} // end main

/\*\*

\* This method prompts for and retrieves a file name from the user and creates it unless it exists.

\* @return Scanner object that refers to the opened input file.

\* @throws IOException

\*/

static Scanner openFile() throws IOException {

Scanner input = new Scanner(System.in);

System.out.print("Enter grade input file in the form filename.ext:");

String filename = input.nextLine();

File file = new File(filename);

if (!file.exists()) {

System.out.println(file.getName()+ " does not exist");

System.exit(1);

}

input = new Scanner(file);

return input;

}

/\*\*

\* @precondition the input file parameter has been successfully opened.

\* @param inFile, Scanner object that refers to the input file.

\* @return array of String which contains the student's name and GPA

\* @throws IOException

\*/

static String [] processGrades(Scanner inFile) throws IOException{

String [] result= new String [3] ;

String studentName = inFile.nextLine();

result[0] = studentName;

float totalQualityPoint = 0;

int totalHours = 0;

while (inFile.hasNextLine()) {

String gradeInfo = inFile.nextLine();

String [] gradeInfoArray= gradeInfo.split(",");

int hours = Integer.parseInt(gradeInfoArray[1]);

int point = 0;

//System.out.println(gradeInfoArray[0] + " " + gradeInfoArray[1]);

switch(gradeInfoArray[0]) {

case "A":

point = 4;

break;

case "B":

point = 3;

break;

case "C":

point = 2;

break;

case "D":

point = 1;

break;

}

totalHours += hours;

totalQualityPoint += (point \* hours);

}

float gpa = totalQualityPoint / totalHours;

DecimalFormat df = new DecimalFormat("0.00");

df.setMaximumFractionDigits(2);

result[1] = df.format(gpa);

return result;

}

/\*\*

\*

\* @param gradeSummary, array of string which contains the student's name and GPA

\* @throws IOException

\* @return void, creates afile with the student's name and stores student's the name and GPA in the file.

\*/

static void storeGpa(String [] gradeSummary) throws IOException {

String filename = gradeSummary[0] + ".csv";

PrintWriter outFile = new PrintWriter(filename);

outFile.println(gradeSummary[0] + "," + gradeSummary[1]);

outFile.close();

}

}

/\*\*

\* This class creates a Rectangle which has a length and width.

\* It also has the methods for calculating the area and perimeter.

\* @author Mehmet ACIKGOZ

\*

\*/

public class Rectangle{

private double length;

private double width;

// default constructor

public Rectangle() {

setLength(1);

setWidth(1);

}

// parameterized constructor

public Rectangle(double length, double width) {

setLength(length);

setWidth(width);

}

// mutators

public void setLength(double length) {

if ( length > 0 )

this.length = length;

}

public void setWidth(double width) {

if ( width > 0 )

this.width = width;

}

// accessors

public double getLength() {

return this.length;

}

public double getWidth() {

return this.width;

}

// other methods

public double getArea() {

return ( getLength()\*getWidth() );

}

public double getPerimeter() {

return 2\*(getLength()+getWidth());

}

}

public class CreateRectangle {

public static void main(String[] args) {

System.out.println("Test rectangle construction:");

Rectangle defaultRec = new Rectangle();

Rectangle square = new Rectangle(4.5, 4.5);

Rectangle errorRec = new Rectangle(-1, -2);

System.out.printf("defaultRec: length = %.2f, width = %.2f%n",

defaultRec.getLength(), defaultRec.getWidth());

System.out.printf("square: length = %.2f, width = %.2f%n",

square.getLength(), square.getWidth());

System.out.printf("errorRec: length = %.2f, width = %.2f%n%n",

errorRec.getLength(), errorRec.getWidth());

System.out.println("Test setting various dimensions:");

defaultRec.setLength(-10);

defaultRec.setWidth(-5);

System.out.printf("defaultRec: length = %.2f, width = %.2f%n",

defaultRec.getLength(), defaultRec.getWidth());

defaultRec.setLength(10);

defaultRec.setWidth(5);

System.out.printf("defaultRec: length = %.2f, width = %.2f%n%n",

defaultRec.getLength(), defaultRec.getWidth());

System.out.println("Test area and perimeter calculations:");

System.out.printf("defaultRec: area = %.2f, perimeter = %.2f%n",

defaultRec.getArea(), defaultRec.getPerimeter());

System.out.printf("square: area = %.2f, perimeter = %.2f%n",

square.getArea(), square.getPerimeter());

}

}

/\*\*

\* This class creates an Item with name, price and quantity

\* and also has the setters and getters as well as reduce method.

\*

\* @author Mehmet ACIKGOZ

\*

\*/

public class Item {

private String name;

private double price;

private int quantity;

public Item() {

this("", 0.0, 0);

}

public Item(String name , double price, int quantity) {

setName(name);

setPrice(price);

setQuantity(quantity);

}

public void setName(String name) {

this.name = name;

}

public void setPrice(double price) {

if (price < 0.0)

price = 0.0;

this.price = price;

}

public void setQuantity(int quantity) {

if (quantity < 0)

quantity = 0;

this.quantity = quantity;

}

public String getName() {

return this.name;

}

public double getPrice() {

return this.price;

}

public int getQuantity() {

return this.quantity;

}

public void reduce (int amount) {

if (amount > 0)

this.quantity -= amount;

}

}

public class CreateItem

{

public static void main(String[] args)

{

System.out.println("Test the default constructor and set methods: ");

Item item1 = new Item();

item1.setName("Computer");

item1.setPrice(500);

item1.setQuantity(10);

System.out.printf("Item: %s, %.2f, %d%n%n",

item1.getName(), item1.getPrice(), item1.getQuantity());

System.out.println("Test a new item and invalid data: ");

Item item2 = new Item("Basketball", -19.99, -10);

System.out.printf("Item: %s, %.2f, %d%n",

item2.getName(), item2.getPrice(), item2.getQuantity());

item2.setPrice(-19.99);

item2.setQuantity(-10);

System.out.printf("Item: %s, %.2f, %d%n",

item2.getName(), item2.getPrice(), item2.getQuantity());

item2.setPrice(19.99);

item2.setQuantity(10);

System.out.printf("Item: %s, %.2f, %d%n%n",

item2.getName(), item2.getPrice(), item2.getQuantity());

System.out.println("Attempt to reduce the number of basketballs by -2");

item2.reduce(-2);

System.out.printf("Item: %s, %.2f, %d%n",

item2.getName(), item2.getPrice(), item2.getQuantity());

System.out.println("Attempt to reduce the number of basketballs by 2");

item2.reduce(2);

System.out.printf("Item: %s, %.2f, %d%n",

item2.getName(), item2.getPrice(), item2.getQuantity());

}

}

/\*\*

\*

\* @author Mehmet ACIKGOZ

\*

\*/

public class FindHighestInMatrix {

/\*\*

\* This program creates a matrix with random numbers between 0 and MAX\_VALUE

\* and finds the heighest number in it and reports it in addition to matrix itself.

\* @param args

\*/

public static void main(String[] args) {

final int ROW = 10;

final int COL = 10;

final int MAX\_VALUE = 99;

int [][] matrix = new int [ROW][COL];

for (int i = 0; i < matrix.length; i++) {

for (int j = 0; j < matrix[i].length; j++) {

// for the inclusive random function: Min + (int)(Math.random() \* ((Max - Min) + 1))

matrix[i][j] = 0 + (int) (Math.random() \* (MAX\_VALUE+1) );

}

}

int theHeighest = matrix[0][0];

int indexRowtheHeighest = 0;

int indexColtheHeighest = 0;

for (int i = 0; i < matrix.length; i++) {

for (int j = 0; j < matrix[i].length; j++) {

if ( matrix[i][j] > theHeighest) {

theHeighest = matrix[i][j];

indexRowtheHeighest = i;

indexColtheHeighest = j;

}

}

}

for (int i = 0; i < matrix.length; i++) {

for (int j = 0; j < matrix[i].length; j++) {

System.out.printf("%3d", matrix[i][j]);

}

System.out.println();

}

System.out.println();

System.out.printf("The highest value located at [%d][%d] is %d" , indexRowtheHeighest, indexColtheHeighest, theHeighest);

}

}

import java.util.Scanner;

import java.io.IOException;

import java.text.DecimalFormat;

/\*\*

\*

\* @author Mehmet ACIKGOZ

\*

\*/

public class KansasPopulationStats {

/\*\*

\* This program reads cities and their population from an input file,

\* finds the Largest city and displays some statistics.

\* @param args

\*/

public static void main(String[] args) throws IOException {

String filename = "D:\\JCCC\\Courses\\\_Spring\_2018\\CS205\\Unit08\\KansasStatistics\\KansasSortedCities.csv";

java.io.File file = new java.io.File(filename);

if ( !file.exists() ) {

System.out.println(file.getName() + " does not exist");

System.exit(1);

}

Scanner input = new Scanner(file);

DecimalFormat decimalFormat = new DecimalFormat("#,###");

int totalPopulation = 0;

String largestCity ="";

int biggestPopulation = 0;

while (input.hasNextLine()) {

String cityInfo = input.nextLine();

String [] cityInfoArray = cityInfo.split(",");

int Population = Integer.parseInt(cityInfoArray[1]);

totalPopulation += Population;

if (Population > biggestPopulation) {

biggestPopulation = Population;

largestCity = cityInfoArray[0];

}

}

System.out.println("Total Population: " + decimalFormat.format(totalPopulation));

System.out.println("Largest City: " + largestCity + " (" + decimalFormat.format(biggestPopulation) + ")" );

input.close();

}

}

import java.io.IOException;

import java.util.Scanner;

/\*\*

\*

\* @author Mehmet ACIKGOZ

\*

\*/

public class CalculateTownPopulation2 {

/\*\*

\* This programs reads the input file and calculates the population of the towns in the file

\* @param args

\*/

public static void main(String[] args) {

// TODO Auto-generated method stub

String filename = "D:\\JCCC\\Courses\\\_Spring\_2018\\CS205\\Unit08\\KansasTownsPop2.txt";

Scanner input = null;

java.io.File file = null;

try {

file = new java.io.File(filename);

/\*

if ( !file.exists() ) {

//System.out.println( file.getName() + " does not exist");

//throw IOException

}

\*/

input = new Scanner(file);

}

catch (IOException ex){

System.out.println (ex.toString());

System.out.println("Could not find file " + file.getName());

System.exit(1);

}

String cityNames = "";

int totalPopulation = 0;

while(input.hasNext()) {

String cityName = input.next();

int Population = input.nextInt();

cityNames = cityNames + " " + cityName;

totalPopulation += Population;

}

System.out.printf("The total population of %s is %d.", cityNames, totalPopulation);

input.close();

}

}

import java.io.IOException;

import java.util.Scanner;

/\*\*

\*

\* @author Mehmet ACIKGOZ

\*

\*/

public class CalculateTownPopulation {

/\*\*

\* This programs reads the input file and calculates the population of the towns in the file

\* @param args

\*/

public static void main(String[] args) throws IOException {

// TODO Auto-generated method stub

String filename = "D:\\JCCC\\Courses\\\_Spring\_2018\\CS205\\Unit08\\KansasTownsPop2.txt";

java.io.File file = new java.io.File(filename);

if ( !file.exists() ) {

System.out.println( file.getName() + " does not exist");

System.exit(1);

}

Scanner input = new Scanner(file);

String cityNames = "";

int totalPopulation = 0;

while(input.hasNext()) {

String cityName = input.next();

int Population = input.nextInt();

cityNames = cityNames + cityName+ " " ;

totalPopulation += Population;

}

System.out.printf("The total population of %s is %d.", cityNames, totalPopulation);

input.close();

}

}

import java.util.Scanner;

/\*\*

\*

\* @author Mehmet ACIKGOZ

\*

\* This program converts any letter to their corresponding phone digits.

\*/

public class ConvertLetterToPhoneNumber {

/\*\*

\* This program converts any letter to their corresponding phone digits.

\* @param args

\*/

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

// Get the phone number

System.out.print("Phone number to convert: ");

String phoneNumber = input.nextLine();

// Process each character in the phone number for display

for (int i = 0; i < phoneNumber.length(); ++i){

// Get the character

char ch = phoneNumber.charAt(i);

if (Character.isLetter(ch)) {

ch = Character.toUpperCase(ch);

System.out.print(getNumber(ch));

}

else {

System.out.print(ch);

}

}

System.out.println();

input.close();

} // end method

/\*\*

\* method getNumber returns the phone digit that corresponds to

\* an upper case letter

\* @param uppercaseLetter An upper case letter to convert

\* @precondition uppercaseLetter is a valid upper case character

\* @return the corresponding digit as an integer

\*/

public static int getNumber(char ch) {

if (ch >= 'A' && ch <= 'C')

return 2;

else if (ch >= 'D' && ch <='F')

return 3;

else if (ch >= 'G' && ch <='I')

return 4;

else if (ch >= 'J' && ch <='L')

return 5;

else if (ch >= 'M' && ch <='O')

return 6;

else if (ch >= 'P' && ch <='S')

return 7;

else if (ch >= 'T' && ch <='V')

return 8;

else

return 9;

}

}

import java.util.Scanner;

/\*\*

\*

\* @author Mehmet ACIKGOZ

\*

\*/

public class PasswordChecker {

/\*\*

\* This program checks password validity.

\* @param args

\*/

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

// Get the password

System.out.print("Password: ");

String password = input.nextLine();

// Check for validity

if (isValidPassword(password))

System.out.println("Valid Password");

else

System.out.println("Invalid Password");

input.close();

}

/\*\*

\* method isValidPassword checks if a password meets the

\* following criteria:

\* 1) >= 8 characters

\* 2) At least 2 digits

\* 3) Consists only of letters and digits

\* @param password The password to check

\* @return true or false depending if the password is valid

\*/

public static boolean isValidPassword(String passwd) {

if (passwd.length() < 8) {

return false;

}

int numOfDigit = 0;

for (int i = 0; i < passwd.length(); i++) {

char character = passwd.charAt(i);

if (Character.isDigit(character) )

numOfDigit++;

}

if (numOfDigit < 2) {

return false;

}

for (int i = 0; i < passwd.length(); i++) {

char character = passwd.charAt(i);

//if ( !Character.isDigit(character) && !Character.isLetter(character) )

if ( !( Character.isDigit(character) || Character.isLetter(character) ) )

return false;

}

return true;

}

}

import java.util.Scanner;

import java.util.Random;

/\*\*

\*

\* @author Mehmet ACIKGOZ

\* This program finds and returns the highest value in an array

\*

\*/

public class FindHighestInArray {

/\*\*

\* This program finds and returns the highest value in an array

\* @param args

\*/

public static void main(String[] args) {

Random rand = new Random();

final int MAX\_SIZE = 10;

int [] arr = new int[MAX\_SIZE];

for (int i = 0; i < MAX\_SIZE; i++) {

// Two different ways for random...

// for the inclusive random function: Min + (int)(Math.random() \* ((Max - Min) + 1))

// arr[i] = -100 + (int )(Math.random() \* ( (100- (-100) ) + 1) );

// for Random

arr[i] = rand.nextInt( 200 + 1) - 100;

}

System.out.println("Max : " + findHighest(arr) );

}

/\*\*

\* This methods find the highest in an array and returns it.

\* @param arr, array of integer

\* @return max value in the array

\*/

public static int findHighest(int [] arr) {

int max = arr[0];

for (int i = 0; i < arr.length; i++) {

if (arr[i] > max) {

max = arr[i];

}

}

return (max);

}

}

import java.util.Scanner;

import java.text.DecimalFormat;

/\*\*

\*

\* @author Mehmet ACIKGOZ

\*

\*/

public class SalaryRaise {

/\*\*

\* This program calculates and gives employee a raise

\* @param args

\*/

public static void main(String[] args) {

double [] salary = {100000, 200000, 300000, 400000, 500000};

DecimalFormat decimalFormat = new DecimalFormat("###,###.00");

Scanner input = new Scanner(System.in);

// Getting the raise from the user

System.out.print("Please enter a raise percentage: ");

double raise = input.nextDouble();

// calculating the new salaries

for (int i = 0; i < salary.length; i++) {

salary[i] \*= ( 1 + raise/100.0 );

}

// printing the salaries

for (double value:salary) {

//System.out.printf("Final salary: $ %.2f", value);

System.out.println(decimalFormat.format(value));

}

}

}

/\*\*

\*

\* @author Mehmet ACIKGOZ

\* This program compute a temperature including the wind chill factor by using a

\* a method called computeWindChill

\*

\*/

public class CalculateWindChill {

/\*\*

\* This program compute a temperature including the wind chill factor

\* @param args

\*/

public static void main(String [] args) {

// for the inclusive random function: Min + (int)(Math.random() \* ((Max - Min) + 1))

int temperature = 0 + (int )(Math.random() \* 50 + 1);

int windSpeed = 4 + (int)(Math.random() \* ((30 - 4) + 1));

double twc = computeWindChill(temperature, windSpeed);

System.out.println("Temperature: " + temperature + " degrees Fahrenheit");

System.out.println("Wind speed: " + windSpeed + " MPH");

System.out.printf("Temperature (including wind chill): %.1f degrees Fahrenheit", twc );

}

/\*\*

\*

\* This method will compute the windchill temperature (float or double) given a

\* temperature and wind speed

\* @pre temperature <= 50 and wind speed > 3

\* @param temperature, in degrees Fahrenheit

\* @param windSpeed, in MPH

\* @return temperature including windchill in degrees Fahrenheit

\*/

public static double computeWindChill(int temperature, int windSpeed) {

double wc = 35.74 + 0.6215 \* temperature - 35.75 \* Math.pow(windSpeed,0.16) +

0.4275 \* temperature \* Math.pow(windSpeed,0.16);

return wc;

}

}

import java.util.Scanner;

/\*\*

\*

\* @author Mehmet ACIKGOZ

\*

\* This program will sum all the integers from a lower limit through an upper limit.

\*/

public class CalculateLoopSum {

/\*\*

\* This program will sum all the integers from a lower limit through an upper limit.

\* It uses an initial loop to ensure a valid upper and lower bound are entered.

\*/

public static void main(String [] args) {

Scanner input = new Scanner(System.in);

int lowerLimit = 0, upperLimit = 0, sum = 0;

do {

System.out.print("Lower limit: ");

lowerLimit = input.nextInt();

System.out.print("Upper Limit: ");

upperLimit = input.nextInt();

System.out.println();

} while (lowerLimit > upperLimit);

for (int i = lowerLimit; i <= upperLimit; i++) {

sum += i;

}

System.out.println("The sum from " + lowerLimit + " to " + upperLimit + " is " + sum);

}

}

import java.util.Scanner;

/\*\*

\*

\* @author Mehmet ACIKGOZ

\* This program will call a method that displays a simple multiplication table

\*

\*/

public class MultiplicationTable {

/\*\*

\* This part of the program is main part of the class that

\* will call a method that displays a simple multiplication table

\* @param none

\*/

public static void main(String [] args) {

Scanner input = new Scanner(System.in);

System.out.print("Number of rows: ");

int row = input.nextInt();

System.out.print("Number of columns: ");

int col = input.nextInt();

System.out.println();

displayTable(row, col);

}

/\*\*

\* This method displays the multiplication table on the screen

\* @param row, number of rows in the table to be displayed

\* @param col, number of columns in the table to be displayed

\* @return void

\*/

public static void displayTable(int row, int col) {

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

for (int j = 1; j <= col; j++) {

//System.out.print( (i\*j) + "\t" );

System.out.printf("%-4d", (i\*j) );

}

System.out.println();

}

}

}

import java.util.Scanner;

/\*\*

\* @author Mehmet ACIKGOZ

\*/

public class Acceleration {

/\*\*

\* Lab 02 Program 2

\* This program finds the acceleration when starting and ending velocities

\* as well as time span are given.

\*/

public static void main(String [] args) {

Scanner input = new Scanner(System.in);

/\*

\* Getting the inputs

\*/

System.out.print("Enter starting velocity in meters / second : ");

double firstVelocity = input.nextDouble();

System.out.print("Enter ending velocity in meters / second : ");

double secondVelocity = input.nextDouble();

System.out.print("Enter the time in seconds : ");

double timeSpan = input.nextDouble();

/\*

\* testing the validity of the inputs and displaying error/result

\*/

if (firstVelocity < 0 || secondVelocity < 0 || timeSpan < 0) {

System.out.println("Invalid data. Please check data entered.");

}

else {

double acceleration = ( secondVelocity - firstVelocity ) / timeSpan;

// The following line will print the result with full precision

// System.out.println("The average acceleration is " + acceleration + " meters / second ");

// The following line will print the result with 3 digits after the decimal

System.out.printf("Acceleration : %.3f meters / second ", acceleration);

}

}

}

import java.util.Scanner;

/\*\*

\*

\* @author Mehhmet ACIKGOZ

\*

\*/

public class DepositCheck {

/\*\*

\* This program will process bank account and prevent a negative balance

\* Unit 5 - Lab

\*/

public static void main(String [] args) {

Scanner input = new Scanner(System.in);

System.out.print("Please enter a balance : $");

double balance = input.nextDouble();

System.out.print("\nPlease enter check amount (-1 to end): $");

double check = input.nextDouble();

while(check > 0) {

if (check > balance) {

System.out.println("Warning: Check will bounce. No transaction occurred.");

}

else {

balance -= check;

System.out.printf("Remaining balance: $%.2f" , balance);

System.out.println();

}

System.out.print("\nPlease enter check amount (-1 to end): $");

check = input.nextDouble();

}

System.out.printf("Final balance: $ %.2f", balance);

}

}

public class Test {

public static void main(String[] args) {

// TODO Auto-generated method stub

int value = (50/100);

System.out.print(value);

}

}

import java.util.Scanner;

/\*\*

\*

\* @author Mehmet ACIKGOZ

\*

\*/

public class Summation {

/\*

\* This program sums ...

\*/

public static void main(String [] args) {

Scanner input = new Scanner(System.in);

int lowerLimit = 0, upperLimit = 0, sum = 0;

do {

System.out.print("Lower limit: ");

lowerLimit = input.nextInt();

System.out.print("Upper Limit: ");

upperLimit = input.nextInt();

System.out.println();

} while (lowerLimit > upperLimit);

for (int i = lowerLimit; i <= upperLimit; i++) {

sum += i;

}

System.out.println("The sum from " + lowerLimit + " to " + upperLimit + " is " + sum);

}

}

import java.util.Scanner;

/\*\*

\* @author Mehmet ACIKGOZ

\*/

public class Project1 {

/\*

\* CS205 - Project 1

\*

\* This program calculates the number of miles per gallon and the number of kilometers per liter when

\* gallons used and starting and ending odometer readings are given.

\* The results are displayed to the user.

\*

\*/

public static void main(String [] args) {

Scanner input = new Scanner(System.in);

final double KM\_PER\_MILE = 1.60934;

final double LITER\_PER\_GALLON = 3.78541;

System.out.print("Please enter number of gallons used: ");

double gallonsUsed = input.nextDouble();

if (gallonsUsed > 0.0) {

System.out.print("Please enter the beginning odometer reading: ");

int startingOdometer = input.nextInt();

System.out.print("Please enter the ending odometer reading: ");

int endingOdometer = input.nextInt();

if (endingOdometer > startingOdometer) {

//System.out.println("");

double milesPerGallons = (endingOdometer - startingOdometer) / gallonsUsed;

double kmPerLiters = milesPerGallons \* KM\_PER\_MILE / LITER\_PER\_GALLON;

System.out.printf("\nThe number of miles per gallons is: %.4f", milesPerGallons);

System.out.printf("\nThe number of kilometers per liter is: %.4f", kmPerLiters );

}

else {

System.out.print("I'm sorry, the ending odometer reading must be greater than the starting odometer reading.");

}

}

else {

System.out.print("I'm sorry, gallons used must be greater than 0.");

}

}

}

public class EscapeSequences {

public static void main(String [] args ) {

System.out.println("\tCapital\t\tState");

System.out.println("\tTopeka\t\tKansas");

}

}

import java.util.Scanner;

/\*\*

\* @author Mehmet ACIKGOZ

\*/

public class AreaOrCircumference {

/\*

\* Lab 03 Program 1

\* This programs displays the area or circumference

\* according to the user input after a radius is entered.

\*

\*/

public static void main(String [] args) {

Scanner input = new Scanner(System.in);

// Getting the input for radius

System.out.print("Please enter the radius: ");

double radius = input.nextDouble();

// Getting the user preference for area or circumference

System.out.print("Enter A (Area) or C (Circumference): ");

String choice = input.next();

// doing the math and displaying either the result or error message

if (choice.equalsIgnoreCase("A")){

double area = Math.PI \*Math.pow(radius,2);

//System.out.print("Area: " + area + " square feet");

System.out.printf("Area: %.4f square feet", area);

}

else if (choice.equalsIgnoreCase("C")){

double circumference= 2.0 \* Math.PI \*radius;

//System.out.print("Circumference: " + circumference + " feet");

System.out.printf("Circumference: %.4f feet", circumference);

}

else

{

System.out.println("I am sorry, I did not understand your choice");

}

}

}

//package test;

import java.util.Scanner;

/\*

@author: Mehmet ACIKGOZ

This program calculates and displays the sum, product, difference,

quotient (division), and remainder of two integers given.

\*/

public class TwoIntegers {

public static void main(String [] args) {

// Getting numbers from the user

System.out.print("Please enter the first positive number : ");

Scanner input = new Scanner(System.in);

int num1 = input.nextInt();

System.out.print("Please enter the second positive number : ");

int num2 = input.nextInt();

// Calculating the results and printing

System.out.println("Sum : " + (num1 + num2) );

System.out.println("Product : " + (num1 \* num2) );

System.out.println("Difference : " + (num1 - num2) );

System.out.println("Quotient : " + (num1 / num2) );

System.out.println("Remainder : " + (num1 % num2) );

}

}

import java.util.Scanner;

/\*\*

\* @author Mehmet ACIKGOZ

\*/

public class MonthAndSeason {

/\*

\* Lab 03 Program 2

\* This program outputs the season when the user enters a numeric month.

\*

\*/

public static void main(String [] args) {

Scanner input = new Scanner(System.in);

// Gettingbthe user input

System.out.print("Enter a numeric month and I will display the season.\nMonth: ");

int month = input.nextInt();

// checking the input and displaying the result.

switch(month) {

case 12:

case 1:

case 2:

System.out.print("winter");

break;

case 3:

case 4:

case 5:

System.out.print("spring");

break;

case 6:

case 7:

case 8:

System.out.print("summer");

break;

case 9:

case 10:

case 11:

System.out.print("fall");

break;

default:

System.out.print("I’m sorry, an invalid month was entered.");

}

}

}

import java.util.Scanner;

/\*

@author: Mehmet ACIKGOZ

This program calculates Body Mass Index when height and weight are given.

\*/

public class BodyMassIndex {

public static void main(String [] args) {

// constants

final double POUND\_TO\_KG = 0.45359237;

final double INCH\_TO\_M = 0.0254;

Scanner input = new Scanner(System.in);

// asking for inputs

System.out.print("Enter weight in pounds: ");

double weight = input.nextDouble();

weight \*= POUND\_TO\_KG ;

System.out.print("Enter height in inches: " );

double height = input.nextDouble();

height \*= INCH\_TO\_M ;

// doing the math

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

// printing the result

System.out.print("BMI is: " + bmi );

}

}

// @author: Mehmet Acikgoz

//package test;

// This programs draws a house

public class Home {

public static void main(String [] args) {

System.out.println(" \* ");

System.out.println(" \* \*");

System.out.println(" \* \*");

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*");

System.out.println("\* \*");

System.out.println("\* \*");

System.out.println("\* \*\*\*\*\* \*");

System.out.println("\* \* \* \*");

System.out.println("\*\*\*\*\* \*\*\*\*\*");

}

}