# ScoreArray.java program

## Question(s)

Create a new class named ScoreArray.java.

This class will have a main() method that does the following:

a. Declares and initializes an array of 8 (8) integers. Name the array appropriately

b. User a decision structure of your choice to ensure input is validated.

i. Only accept numeric data in the form of an integer

ii. The acceptable range is 0 to 100 i inclusive.

iii. If user input is not valid, display an appropriate custom error message and re-prompt the user for valid data. iv. Hint: use a loop. No processing should occur if data is invalid.

c. Declare numeric variables named averageScore, highestScore, lowestScore that is initially set to zero (0). These will hold the calculations for averages, highest, and lowest scores.

d. Create a for loop that gets the length of the score array as the condition for continued looping. This for loop should display the score number and its value (e.g. “Score 1 = 96”).

e. As well, each loop through should add the score to the average variable.

f. You also need to determine and display the highest and lowest score in the array.

g. After the loop finishes executing, display the average score, and highest score, and the lowest score. Note: do not use the literal value (i.e. the number 8) as the denominator in the average calculation, best practice is to determine the length of the array.

## Code

/\*  
 Author: Christian Powlette  
 Due Date: February 16, 2024  
 Professor: Yuvaraj Sambandan  
 Course: COSC-1200-06 Object-Oriented Programming  
 Description of class: Program that calculated the highest, lowest, and average of user input  
\*/  
  
import java.util.Scanner;  
  
public class ScoreArray {  
 public static void main(String[] args) {  
  
 // DECLARATIONS  
 int[] scores = new int[8]; // Declare and initialize an array of 8 integers  
 Scanner myObj = new Scanner(System.*in*); // Create a Scanner object for user input  
 double averageScore = 0; // Initialize double averageScore to calculate average later on  
 int highestScore = Integer.*MIN\_VALUE*; // Initialize with the smallest possible value  
 int lowestScore = Integer.*MAX\_VALUE*; // Initialize with the largest possible value  
 int scoresInput; // Initialize the variable for input  
  
 // MAIN PROGRAM  
 for (int i = 0; i < scores.length; i++) {  
  
 // Loop until valid input is received  
  
 System.*out*.print("Enter numeric integer score " + (i + 1) + ": ");  
 try {scoresInput = Integer.*parseInt*(myObj.next()); // Check that input received is a numeric integer  
 if (scoresInput >= 0 && scoresInput <= 100) {  
 scores[i] = scoresInput; // Store each value in the array during each loop  
 averageScore += scores[i]; // Update the averageScore  
 highestScore = Math.*max*(highestScore, scores[i]); // Update the highestScore  
 lowestScore = Math.*min*(lowestScore, scores[i]); // Update the lowestScore  
 break; // Exit the loop if valid input  
 } else {  
 System.*out*.println("Please enter a numeric integer between 0 and 100.");  
 }  
 } catch (Exception e) {  
 System.*out*.println("Invalid input. Please enter a numeric integer.");  
 }  
 }  
  
  
 // Calculate the average  
 averageScore /= scores.length;  
  
 // Display results  
 System.*out*.println("Highest score: " + highestScore);  
 System.*out*.println("Lowest score: " + lowestScore);  
 System.*out*.println("Average score: " + averageScore);  
 }  
}

## Output

Enter numeric integer scores 1: 10

Enter numeric integer scores 2: 20

Enter numeric integer scores 3: 30

Enter numeric integer scores 4: asas

Error! Input is not numeric Integer

Enter numeric integer scores 4: 40

Enter numeric integer scores 5: 50

Enter numeric integer scores 6: 24

Enter numeric integer scores 7: 32

Enter numeric integer scores 8: 27

Score 1 = 10

Score 2 = 20

Score 3 = 30

Score 4 = 40

Score 5 = 50

Score 6 = 24

Score 7 = 32

Score 8 = 27

Average score: 3.9353183060884476

Highest score: 50

Lowest score: 10

# PlayingWithStrings.java

## Question(s)

Create a new class named PlayingWithStrings.java.

This class will have a main() method that does the following:

a. Instantiates and initializes a String object named lastName (should be initialize to YOUR LAST NAME)

b. Display in a sentence the length of your last name (using the appropriate String method)

c. Declare an integer named thirdLastLetter that is equal to 3 less than the length of your last name. d. Use an appropriate String method to display the third-to-last character of your last name (in a complete sentence).

e. Use an appropriate String method to display your last name all-in upper-case letters.

f. Use an appropriate String method to display your last name all-in lower-case letters.

g. Create an if statement that compares your last name to the string “Short”, and displays whether or not you have the same last name as your professor. Add to that if statement an else if that compares your last name to the string “Patel”, and displays whether or not you have the same last name as a few students in our class.

h. Create an if statement that determines whether your last name contains the character sequence “ne”. If your last name contains “ne” a statement should state that your last name (display it) contains the character sequence “ne”, or it should say that it does not. e.g. Display for the instructor will be: The last name Short does not contain “ne”

i. Please take notice of the double quotes. Note: Google the Java API to find which method checks for character sequences in a String object.

## Code

/\*  
 Author: Christian Powlette  
 Due Date: February 16, 2024  
 Professor: Yuvaraj Sambandan  
 Course: COSC-1200-06 Object-Oriented Programming  
 Description of class: A program Displays a multitude of String Methods  
\*/  
public class PlayingWithStrings {  
 public static void main(String[] args) {  
  
 //DECLARATIONS  
 String lastname = "Powlette";  
 int thirdLastLetter = lastname.length() - 3;  
  
 //MAIN PROGRAM  
 System.*out*.println("My lastname '" + lastname + "' is " + lastname.length() + " characters long.");  
 System.*out*.println("The third-to-last character of my last name is '" + lastname.charAt(thirdLastLetter) + "'.");  
 System.*out*.println("My last name in upper-case is '" + lastname.toUpperCase() + "'.");  
 System.*out*.println("My last name in lower-case is '" + lastname.toLowerCase() + "'.");  
 if (lastname == "Short") {  
 System.*out*.println("I have the same last name as Professor Short.");  
 }else if (lastname != "Short") {  
 System.*out*.println("My lastname is not the same as Professor Short");  
 }if (lastname == "Patel") {  
 System.*out*.println("My last name is the same as 'Patel'");  
 } else if (lastname != "Patel") {  
 System.*out*.println("MY last name is not the same as 'Patel'");  
 }if (lastname.contains("ne")) {  
 System.*out*.println("My lastname, " + lastname + " contains the character sequence 'ne'");  
 }else if (!lastname.contains("ne")) {  
 System.*out*.println("My lastname, " + lastname + " does not contain the character sequence 'ne'");  
 }  
  
 }  
}

## Output

My lastname 'Powlette' is 8 characters long.

The third-to-last character of my last name is 't'.

My last name in upper-case is 'POWLETTE'.

My last name in lower-case is 'powlette'.

My lastname is not the same as Professor Short

# FunWithMath.java

## Question(s)

Create a new class named FunWithMath.java. For this program you will be required to use the Math class (found in the java.lang package) and the DecimalFormat class and the NumberFormat.getCurrencyInstance() method (these are found in the java.text package and will need to be imported). Inside the main() method, do the following:

a. This program will calculate the present value of a future amount, using the compound interest formula:

presentValue = futureValue/(1 + annualInterestRate)yearsInFuture

b. Declare primitives (and initialize them) for the following (choose appropriate data types) : i. currentValue = 0 ii. futureValue = 1000000 iii. annualInterestRate = this should be a random number between 0% and 10% (use the Math.random() method) iv. yearsInTheFuture = 20

c. Your program should calculate the present value of the one million dollars.

d. Also, instantiate two (2) different (appropriately named) DecimalFormat objects, one that will display the dollar amounts appropriately, and one that will display interest rates to 3 decimal places (a leading zero if the interest rate is less than 1%).

## Code

/\*  
 Author: Christian Powlette  
 Due Date: February 16, 2024  
 Professor: Yuvaraj Sambandan  
 Course: COSC-1200-06 Object-Oriented Programming  
 Description of class: A program that accepts calculates various interest values  
\*/  
import java.text.DecimalFormat;  
  
public class scratch\_4 {  
 public static void main(String[] args) {  
 //DECLARATIONS  
 int currentValue = 0;  
 double futureValue = 1000000;  
 double annualInterestRate = Math.*random*() / 10; // Random interest rate between 0% and 10%  
 int yearsInTheFuture = 20;  
  
 System.*out*.println("Hello");  
 //MAIN PROGRAM  
  
 // Calculate present value using compound interest formula of a future amount PV = FV / (1 + r)^t  
 double presentValue = futureValue /Math.*pow*((1 /+ annualInterestRate),yearsInTheFuture); // PV= 1000000 / (1 + interest) ^ 20  
  
 // Initiate DecimalFormat objects  
 DecimalFormat dollarFormat = new DecimalFormat("$#,###.##"); //formats the Future Value  
 DecimalFormat interestRateFormat = new DecimalFormat("0.000%"); //formats the annual interest rate  
  
 // Display results  
 System.*out*.println("The Future Value is: " + dollarFormat.format(futureValue)); //displays a concatenated future value message  
 System.*out*.println("The Annual Interest Rate is: " + interestRateFormat.format(annualInterestRate)); //displays a concatenated Annual Interest message  
 System.*out*.println("The Years in the Future are: " + yearsInTheFuture); //displays the concatenated time years variable message  
 System.*out*.println("The Present Value is: " + dollarFormat.format(presentValue)); //displays a concatenated present value message  
 }  
}

## Output

The Future Value is: $1,000,000

The Annual Interest Rate is: 2.778%

The Years in the Future are: 20

The Present Value is: $0

# Temp.java

## Question(s)

Create a new class named Temp.java. This class will have a main() method that will:

a. Prompt the user for and accept user input a temperature in the Fahrenheit temperature scale.

b. User a decision structure of your choice to ensure input is validated.

i. Only accept numeric data in the form of a real number

ii. The acceptable range is -130.0 F and 135.0 F inclusive.

iii. If user input is not valid, display an appropriate custom error message and re-prompt the user for valid data.

iv. Hint: use a loop. No processing should occur if data is invalid.

c. Once input has been validated, pass the value to a function you need to create named TempConversion() that will accept a double or float data type (your choice). This function will calculate the temperature conversion and return the calculated value.

d. Display the output that includes the original user input and the converted Celsius value.

## Code

/\*  
 Author: Christian Powlette  
 Due Date: February 16, 2024  
 Professor: Yuvaraj Sambandan  
 Course: COSC-1200-06 Object-Oriented Programming  
 Description of class: A program that calculates the Celsius value when Fahrenheit input is given.  
\*/  
  
import java.text.DecimalFormat;  
import java.util.Scanner;  
  
public class scratch\_2 {  
 public static void main(String[] args) {  
 //DECLARATIONS  
 double Fahrenheit = 0; //  
 Scanner myObj = new Scanner(System.*in*);  
 int process = 0;  
 double user\_input = 0;  
 DecimalFormat degreesFormat = new DecimalFormat("##0.00"); //formats the returned Celsius conversion  
  
 //MAIN PROGRAM  
 while (process == 0) { //loops until input is received in Fahrenheit  
 double Celsius = 0;  
 System.*out*.println("Enter a numeric double value for Celsius in the range of -135.0 to 135.0 to convert to Fahrenheit");  
 try { //try catch statement to only obtain numeric float input  
 Fahrenheit = Double.*parseDouble*(myObj.next()); // obtains input from scanner  
 if (Fahrenheit >= -135.0F && Fahrenheit <= 135.0F) { //check afterward that the input is withing the range of 0 to 100.  
 Celsius = *TempConversion*(Fahrenheit); //store each value in the array during each loop  
 process = 1;  
 }  
 else { //if the input is a numeric float, but does not meet the range specified in the if statement  
 System.*out*.println("Error! Input is larger than 135.0F and less than -135.0F"); // print and Error message saying input is not in range  
 }  
 } catch (Exception e) { //catch exception for errors  
 System.*out*.println("Error! Input is not a numeric Integer."); // prints error message that input is not a numeric integer  
 Fahrenheit = 0.00;  
 continue;  
 }  
 System.*out*.println(Fahrenheit + "°F Fahrenheit is equal to " + degreesFormat.format(Celsius)+ "°C when converted to Celsius.");  
 //System.out.println(Celsius);  
 }  
 }  
  
 //Method that obtains verified input and passes it through function that converts the Celsius input and  
 // converts it into Fahrenheit  
 public static double TempConversion(double num) {//creates method for calculating conversion, only accepts float  
 // Code to be executed  
 double Celsius; //initializes variable  
 //initializes constant double of the "ratio of change" between Fahrenheit and Celsius, needed in conversion formula  
 final double RATIOOFCHANGE = (float) 5 / 9;  
 //initializes constant the for Fahrenheit's freeing point, need for formula  
 final int FAHRENHEITFREEZINGPOINT = 32;  
  
 //Fahrenheit to Celsius Calculation  
 Celsius = RATIOOFCHANGE \* (num - FAHRENHEITFREEZINGPOINT);  
 //Displays output of conversion of Celsius and initial user input for Fahrenheit  
 return Celsius;  
 }  
  
}

## Output

Enter a numeric double value for Celsius in the range of -135.0 to 135.0 to convert to Fahrenheit

abc

**Error! Input is not a numeric Integer.**

Enter a numeric double value for Celsius in the range of -135.0 to 135.0 to convert to Fahrenheit

-137

**Error! Input is larger than 135.0F or less than -135.0F**

Enter a numeric double value for Celsius in the range of -135.0 to 135.0 to convert to Fahrenheit

139

**Error! Input is larger than 135.0F or less than -135.0F**

Enter a numeric double value for Celsius in the range of -135.0 to 135.0 to convert to Fahrenheit

23

23.0°F Fahrenheit is equal to -5.00°C when converted to Celsius.