**Dhanushka Galapitage**

**Email :** [**dakumara24@gmail.com**](mailto:dakumara24@gmail.com)

**Mobile : 0410004692**

**Programming Skills Solutions:**

**01.Language Diversity :**

**Java: CalculatePower.java [with for loop]**

**package** com.question.one;

**import** java.util.InputMismatchException;

**import** java.util.Scanner;

/\*\*

\* **@author** DHANUSHKA

\* This class finds the value of the Number raised to the Power

\*/

**public** **class** CalculatePower {

// Main methods starts here

**public** **static** **void** main(String[] args) {

// Create Scanner object to get the inputs from user

Scanner sc = **new** Scanner(System.***in***);

// Starts try block to catch any exceptions.

**try** {

// Insert base value

System.***out***.println("Please insert base value : ");

**int** baseVal = sc.nextInt(); // Read base value

// Insert exponent value

System.***out***.println("Please insert exponent value : ");

**int** expoVal = sc.nextInt(); // Read exponent value

**int** tempVal; // Declare temporary value

// If exponent is 0, the answer should be 1.

**if** (expoVal == 0) {

tempVal = 1;

} **else** {

tempVal = baseVal; // Assign vase value to the temporary value

// With the for loop, increase and multiple base and exponent values

**for** (**int** i = 1; i < expoVal; i++) {

tempVal = tempVal \* baseVal;

}

}

// Print result value

System.***out***.println("\nPower of the inserted number is : " + tempVal);

// If user input invalid value, exception will be occurred

} **catch** (InputMismatchException e) {

System.***out***.println("Inserted value is not a number...!");

}

// Finally, close the scanner object

sc.close();

}

}

**Java: CalculatePower\_2.java [with while loop]**

**package** com.question.one;

**import** java.util.InputMismatchException;

**import** java.util.Scanner;

/\*\*

\* **@author** DHANUSHKA

\* THis class finds the value of the Number raised to the Power

\*

\*/

**public** **class** CalculatePower\_2 {

**public** **static** **void** main(String[] args) {

// Create Scanner object to get the inputs from user

Scanner sc = **new** Scanner(System.***in***);

// Starts try block to catch any exceptions.

**try** {

// Insert base value

System.***out***.println("Please insert base value : ");

**int** baseVal = sc.nextInt(); // Read base value

// Insert exponent value

System.***out***.println("Please insert exponent value : ");

**int** expoVal = sc.nextInt(); // Read exponent value

// Print result value

System.***out***.println("\nPower of the inserted number is : " + *getPowerValue*(baseVal, expoVal));

// If user input invalid value, exception will be occurred

} **catch** (InputMismatchException e) {

System.***out***.println("Inserted value is not a number...!");

}

// Finally, close the scanner object

sc.close();

}

/\*\*

\* This method finds the value of the Number raised to the Power

\*/

**private** **static** **int** getPowerValue(**int** baseVal, **int** expoVal) {

**int** returnVal = 1;

// With while loop, find to the power value.

**while** (expoVal != 0) {

returnVal \*= baseVal;

--expoVal;

}

**return** returnVal;

}

}

**JavaScript, HTML:** **CalculatePower.html**

<!DOCTYPE html>

<html>

<body>

<p>This class finds the value of the Number raised to the Power</p>

<!-- Call to the calculate power function -->

<button onclick="calculatePower()">Click to find power</button>

<!-- Print the value of result. -->

<p id="demo"></p>

<!-- Javascript function. -->

<script>

function calculatePower() {

// Get the values of base and power from user

var baseValue = prompt("Please enter base value : ", "Integer");

var powerValue = prompt("Please enter power value : ", "Integer");

// validate inputs.

if(isNaN(baseValue) || isNaN(powerValue)){

document.write(" Invalid input, Try again <br/>");

} else{

var i, sum = 1; // Define temporary variables

for (i=0;i<powerValue;i++) { //loops power value times

sum = sum \* baseValue;

}

// Print the result.

document.getElementById("demo").innerHTML = "Power of the inserted number is : " + sum;

}

}

</script>

</body>

</html>

**Python:** **CalculatePower.py**

#Auuthor : Dhanushka

#This program finds the value of the Number raised to the Power

def main():

print("Finding power....")

# Prompt user to enter the base value.

baseValue = input("\nPlease insert base value integer.. : ")

# Prompt user to enter the power value.

powerValue = input("\nPlease insert power value integer.. : ")

# Check inputs are numbers

if(baseValue.isnumeric() and powerValue.isnumeric()):

# Convert strings into integers

baseValue = int(baseValue)

powerValue = int(powerValue)

# Define temporary variable power

power = 1

# Find the power value using for loop.

for i in range(1, powerValue + 1):

power = power \* baseValue

print("\nPower of the inserted number is : ",power)

# If inputs are not numbers, prompt error message.

else:

print("Invalid input. Try again...!")

# Redirect to main method.

main()

**02. Mathematics:**

**Java : Quadratic.java**

**package** com.question.two;

**import** java.util.Scanner;

/\*\*

\* **@author** DHANUSHKA

\* This class solve the Quadratic equation.

\*/

**public** **class** Quadratic {

// Main method starts here.

**public** **static** **void** main(String[] args) {

// Get the inputs from user

Scanner sc = **new** Scanner(System.***in***);

**double** a, b, c;

System.***out***.println("Enter the values for quadratic equation :");

// Get values for 'a', 'b' and 'c'

**try** {

System.***out***.print("a : ");

a = Double.*parseDouble*(sc.nextLine());

System.***out***.print("b : ");

b = Double.*parseDouble*(sc.nextLine());

System.***out***.print("c : ");

c = Double.*parseDouble*(sc.nextLine());

// If invalid input, catch the exception and exit.

} **catch** (Exception e) {

System.***out***.println("Invalid input, Try again");

**return**;

}

// Create determinant value.

**double** determinantVal = Math.*pow*(b, 2) - 4 \* a \* c;

**if** (determinantVal == 0) { // If determinant value is zero.

System.***out***.println("Root value is : " + -b / (2 \* a));

} **else** **if** (determinantVal > 0) { // If determinant value greater than zero.

System.***out***.println("Root values are " + (-b + Math.*sqrt*(determinantVal)) / (2 \* a) + " and "

+ (-b - Math.*sqrt*(determinantVal)) / (2 \* a));

} **else** {

System.***out***.println("No real root values");

}

// Closing scanner object

sc.close();

}

}

**03. Prices and Percentages**

**Java : CalculateTax.java**

**package** com.question.three;

**import** java.util.Scanner;

/\*\*

\* **@author** DHANUSHKA

\* This class determines the original price from selling price and percentage.

\*/

**public** **class** CalculateTax {

**public** **static** **void** main(String[] args) {

// Create scanner object

Scanner sc = **new** Scanner(System.***in***);

System.***out***.println("Determine original price...");

**try** {

// Get the inputs from user

System.***out***.println("Insert tax inclusive price : ");

**double** priceWithTax = sc.nextDouble();

System.***out***.println("Insert tax percentage : ");

**double** taxPercentage = sc.nextDouble();

// Call the method findTaxExclusivePrice

**double** originalPrice = *findTaxExclusivePrice*(priceWithTax, taxPercentage);

System.***out***.println("Original Price is : " + originalPrice);

} **catch** (Exception e) {

System.***out***.println("Invalid input, Try again");

**return**;

}

sc.close();

}

/\*\*

\* This method calculate Original price from selling price and percentage.

\* **@param** priceWithTax, taxPercentage

\* **@return** original price

\*/

**private** **static** **double** findTaxExclusivePrice(**double** priceWithTax, **double** taxPercentage) {

**return** (priceWithTax \* 100) / (100 + taxPercentage);

}

}

**04. String Manipulation**

**Java : Soundex.java**

**package** com.question.four;

/\*\*

\* **@author** DHANUSHKA

\* This class resolves the soundex algorithms

\*/

**public** **class** Soundex {

/\*\*

\* This method gets the surname from user and return soundex code

\* **@param** str

\* **@return** soundex code

\*/

**public** **static** String getSoundexCode(String str) {

// Make the string upper case and create char array

**char**[] arr = str.toUpperCase().toCharArray();

// Assign first character

**char** firstChar = arr[0];

// implement RULE NO 1 (replace letters with numbers)

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

**switch** (arr[i]) {

**case** 'B':

**case** 'F':

**case** 'P':

**case** 'V': {

arr[i] = '1';

**break**;

}

**case** 'C':

**case** 'G':

**case** 'J':

**case** 'K':

**case** 'Q':

**case** 'S':

**case** 'X':

**case** 'Z': {

arr[i] = '2';

**break**;

}

**case** 'D':

**case** 'T': {

arr[i] = '3';

**break**;

}

**case** 'L': {

arr[i] = '4';

**break**;

}

**case** 'M':

**case** 'N': {

arr[i] = '5';

**break**;

}

**case** 'R': {

arr[i] = '6';

**break**;

}

**default**:

arr[i] = '0';

**break**;

}

}

// Convert char letter to string.

String returnCode = Character.*toString*(firstChar);

// implement RULE NO 2 and Rule no 3

// Identify identical digits

**for** (**int** j = 1; j < arr.length; j++) {

// Handle array index out of bound exception

**if** (j > 1) {

// Check digit, '0' or same adjacent digits or '0' with two same

// digits in sides.

**if** (arr[j] != '0' && arr[j - 1] != arr[j] && !(arr[j] == arr[j - 2] && arr[j - 1] == '0')) {

returnCode += arr[j];

}

} **else** {

**if** (arr[j] != '0' && arr[j - 1] != arr[j]) {

returnCode += arr[j];

}

}

}

// implement RULE NO 4(return the correct code)

// Append four '0' s to the end.

returnCode = returnCode.concat("0000");

// Return first four digits.

**return** returnCode.substring(0, 4);

}

}

**SoundexImpl.java**

**package** com.question.four;

**import** java.util.Scanner;

/\*\*

\* **@author** DHANUSHKA

\* This is Soundex Algorithm implementation class.

\* This class gets the surname from user and print the code.

\*/

**public** **class** SoundexImpl {

// Main method

**public** **static** **void** main(String[] args) {

// Get the inputs from user

Scanner sc = **new** Scanner(System.***in***);

System.***out***.print("Insert Surname : ");

String surname = sc.nextLine();

// Call getSoundexCode method from Soundex class.

String soundexCode = Soundex.*getSoundexCode*(surname);

System.***out***.println("\nSoundex code for '" + surname + "' is '" + soundexCode + "'.");

sc.close();

}

}

**05. C Skills**

**Not done**

**06. SQL Skills**

Select CONCAT\_WS(" ", pe.GivenName, pe.FamilyName) AS Person\_Name,

YEAR(curdate()) - YEAR(pe.DateOfBirth) AS Age,

pa.FatherName AS Father\_Name, pa.MotherName AS Mother\_Name FROM person AS pe

LEFT JOIN personparent AS pp ON pe.PersonID = pp.PersonID

LEFT JOIN parent AS pa ON pp.ParentID = pa.ParentID;

**07. Code Review and Critiquing**

Unfortunately, I was not able to finish this part and if I get a chance from your end to do it and submit bit late, please let me know.