**Q1. Process a coffee order: take customer size choice, calculate total price based on size and add-ons, and handle a list of 5 drink types.**

**Objective:**  
Take customer’s coffee order by selecting drink type, size, and add-ons, then calculate and display the total price.

**Key Concepts Used:**

* **Arrays:** Store drink types and add-ons.
* **User Input:** Use Scanner to get choices from user.
* **Switch Statement:** Determine price based on size.
* **String Manipulation:** Handle input and capitalize output.
* **Lists:** Store multiple add-ons selected by user.
* **Parsing & Validation:** Parse add-on inputs and validate choices.
* **Basic Arithmetic:** Calculate total price using size and add-on prices.

**Code:**

import java.util.Scanner; public class CoffeeOrder {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

String[] drinks = {"Latte", "Espresso", "Cappuccino", "Mocha", "Americano"}; System.out.println("Choose drink (0-4): "); int drinkChoice = sc.nextInt();

System.out.println("Choose size (S/M/L): "); char size = sc.next().charAt(0);

double price = 0; switch (size) { case 'S': price = 2.0; break; case 'M': price = 3.0; break; case 'L': price = 4.0; break;

}

System.out.println("Add whipped cream? (yes/no): "); String addOn = sc.next(); if (addOn.equalsIgnoreCase("yes")) {

price += 0.5;

}

System.out.println("You ordered a " + size + " " + drinks[drinkChoice] + ". Total: $" + price);

}

}

**Q2. Create a method that accepts two numbers and an operation symbol. Use a switch to perform and return the result of addition, subtraction, multiplication, or division.**

**Objective**

* Create a method to perform basic arithmetic operations (+, -, \*, /) on two numbers based on an operator symbol.

Key Concepts Used

* Methods: Encapsulate calculation logic in a reusable method calculate().
* Switch Statement: Select operation based on the operator character.
* Exception Handling: Handle division by zero and invalid operators using exceptions.
* Basic Arithmetic Operators: Perform addition, subtraction, multiplication, and division.
* Input/Output: Demonstrate method usage with print statements in main().

**Code:**

import java.util.Scanner; public class SimpleCalculator { public static void main(String[] args) {

Scanner sc = new Scanner(System.in); System.out.print("Enter first number: "); double a = sc.nextDouble();

System.out.print("Enter second number: "); double b = sc.nextDouble();

System.out.print("Enter operation (+, -, \*, /): "); char op = sc.next().charAt(0);

double result = 0; switch (op) { case '+': result = a + b; break; case '-': result = a - b; break; case '\*': result = a \* b; break; case '/': result = b != 0 ? a / b : 0; break; default: System.out.println("Invalid operation");

}

System.out.println("Result: " + result);

}

}

**Q3. Input a string and count vowels, consonants, digits, and special characters using loops and conditionals.**

**Objective**

* Input a string and count the number of vowels, consonants, digits, and special characters.

**Key Concepts Used**

* **Loops:** Traverse each character in the string.
* **Conditionals:** Check character type (vowel, consonant, digit, special).
* **Character Class Methods:** Use Character.isLetter(), isDigit(), and isWhitespace() for classification.
* **String Manipulation:** Convert input to lowercase for uniform checking.
* **Helper Method:** Separate vowel-check logic in isVowel().
* **Input Handling:** Use Scanner to read user input.

**Code:**

import java.util.Scanner; public class CountCharacters { public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

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

String str = sc.nextLine();

int vowels = 0, consonants = 0, digits = 0, special = 0; str = str.toLowerCase();

for (char ch : str.toCharArray()) { if (Character.isDigit(ch)) digits++; else if ("aeiou".indexOf(ch) != -1) vowels++; else if (ch >= 'a' && ch <= 'z') consonants++; else special++;

}

System.out.println("Vowels: " + vowels);

System.out.println("Consonants: " + consonants);

System.out.println("Digits: " + digits);

System.out.println("Special Characters: " + special);

}

}

**Q4. For n customers, input name, account type, and balance. Apply 4% interest for savings and 6% for fixed accounts, then display updated balances.**

**Objective**

* Input details for **n** customers: name, account type, and balance.
* Apply interest (4% for savings, 6% for fixed).
* Display updated balances for all customers.

**Key Concepts Used**

* **Classes/Objects:** Customer class stores customer data and behavior.
* **Encapsulation:** Group related data and methods inside Customer.
* **Constructor:** Initialize customer details.
* **Methods:** Apply interest and display info.
* **Input Handling:** Use Scanner to read data.
* **Validation:** Check valid account types, skip invalid entries.
* **Loops:** Input multiple customers and display results.
* **String Handling:** Use toLowerCase() and capitalization for consistency.
* **Array of Objects:** Store multiple customers.
* **Formatted Output:** Print balances with 2 decimals.

**Code:**

import java.util.Scanner; public class InterestCalculator { public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.print("Enter name: ");

String name = sc.nextLine();

System.out.print("Enter account type (Savings/Fixed): ");

String type = sc.nextLine();

System.out.print("Enter balance: "); double balance = sc.nextDouble(); double rate = type.equalsIgnoreCase("Savings") ? 0.04 : 0.06;

double interest = balance \* rate; double newBalance = balance + interest;

System.out.println("Hello " + name + ", your updated balance is: " + newBalance);

}

}

**Q5. Read 5 daily temperatures into an array. Use a loop and a method to convert each temperature from Celsius to Fahrenheit, displaying both.**

**Objective**

* Read 5 daily temperatures in Celsius from the user.
* Convert each Celsius temperature to Fahrenheit.
* Display both Celsius and Fahrenheit values.

**Key Concepts Used**

* **Arrays:** Store multiple temperature values.
* **Loops:** Input temperatures and display results.
* **Methods:** celsiusToFahrenheit() performs temperature conversion.
* **Input Handling:** Using Scanner to read user input.
* **Arithmetic Operations:** Formula to convert Celsius to Fahrenheit.
* **Formatted Output:** Display temperatures with 2 decimal precision.

**Code:**

import java.util.Scanner; public class TempConverter { public static double celsiusToFahrenheit(double celsius) { return (celsius \* 9 / 5) + 32;

}

public static void main(String[] args) { Scanner scanner = new Scanner(System.in); final int DAYS = 5; double[] celsiusTemps = new double[DAYS];

System.out.println("Enter 5 daily temperatures in Celsius (C):");

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

System.out.print("Day " + (i + 1) + " Temp (C): "); celsiusTemps[i] = scanner.nextDouble();

}

System.out.println("\n--- Temperature Conversions ---"); for (int i = 0; i < DAYS; i++) { double celsius = celsiusTemps[i]; double fahrenheit = celsiusToFahrenheit(celsius);

System.out.printf("Day %d: %.1f C is %.1f F%n",

(i + 1), celsius, fahrenheit);

}

}

}

**Q6. Accept number of units consumed and calculate bill based on slab rates using conditionals and methods.**

**Objective**

* Accept the number of electricity units consumed.
* Calculate the bill using slab rates with different prices for unit ranges.
* Display the total bill amount.

**Key Concepts Used**

* **Conditionals:** if-else to apply slab rates based on units consumed.
* **Methods:** calculateBill() computes the bill based on input units.
* **Input Handling:** Use Scanner to get user input.
* **Arithmetic:** Multiplying units by rate per slab to calculate charges.
* **Validation:** Check for invalid (negative) input.
* **Formatted Output:** Display bill amount with two decimals.

**Code:**

import java.util.Scanner; public class ElectricityBill { public static double calculateBill(int units) { double totalBill = 0.0; if (units <= 100) { totalBill = units \* 2.00;

} else if (units <= 200) { totalBill += 100 \* 2.00; totalBill += (units - 100) \* 3.50;

} else { totalBill += 100 \* 2.00; totalBill += 100 \* 3.50; totalBill += (units - 200) \* 5.00;

}

return totalBill;

}

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.println("Enter the number of units consumed:"); int consumedUnits = scanner.nextInt();

if (consumedUnits < 0) {

System.out.println("Units consumed cannot be negative.");

} else {

double finalBill = calculateBill(consumedUnits); System.out.printf("Electricity Bill for %d units: $%.2f%n", consumedUnits, finalBill);

}

}

}

**Q7. Input a string and check if it’s a palindrome (ignore case and spaces). Use string methods and exception handling.**

**Objective**

* Input a string from the user.
* Check if the string is a palindrome (ignore case and spaces).
* Handle invalid or empty input gracefully.
* Display the result.

**Key Concepts Used**

* **String Manipulation:** Remove spaces and change case for uniform comparison.
* **String Methods:** Use replaceAll(), toLowerCase(), and StringBuilder.reverse().
* **Exception Handling:** Handle empty input or null values using try-catch.
* **Conditionals:** Compare cleaned string with its reverse to determine palindrome.
* **Input Handling:** Use Scanner for user input.

**Code:**

import java.util.Scanner; public class PalindromeChecker { public static boolean isPalindrome(String s) throws IllegalArgumentException { if (s == null || s.trim().isEmpty()) { throw new IllegalArgumentException("Input string cannot be empty or null.");

}

String cleanedString = s.replaceAll("\\s+", "").toLowerCase();

String reversedString = new StringBuilder(cleanedString).reverse().toString();

return cleanedString.equals(reversedString);

}

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.println("Enter a string to check if it's a palindrome:"); String input = scanner.nextLine();

try {

if (isPalindrome(input)) {

System.out.println(" '" + input + "' IS a palindrome.");

} else {

System.out.println(" '" + input + "' IS NOT a palindrome.");

}

} catch (IllegalArgumentException e) {

System.err.println("Error: " + e.getMessage());

}

}

}

**Q8. Read a word (String). Use a loop and a switch on each character to replace 'a' with '4', 'e' with '3', and 'o' with '0'.**

**Objective**

* Read a word from the user.
* Replace letters 'a', 'e', and 'o' with '4', '3', and '0' respectively.
* Display the modified word.

**Key Concepts Used**

* **Loops:** Iterate over each character of the string.
* **Switch Statement:** Decide which characters to replace.
* **StringBuilder:** Efficiently build the modified string.
* **Character Handling:** Use Character.toLowerCase() for case-insensitive replacement.
* **Input Handling:** Use Scanner to read user input.

**Code:**

import java.util.Scanner; public class ReplaceChars { public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

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

String word = sc.nextLine();

StringBuilder result = new StringBuilder();

for (char ch : word.toCharArray()) { switch (ch) { case 'a': result.append('4'); break; case 'e': result.append('3'); break; case 'o': result.append('0'); break; default: result.append(ch);

}

}

System.out.println("Modified word: " + result);

}

}