**Exercise 01:**

Write a Java program that takes an integer as input and determines whether it is even or odd. Use conditional statements to make this determination and print an appropriate message.

Solution:

import java.util.Scanner;

public class EvenOddChecker {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter an integer: ");

int number = scanner.nextInt();

if (number % 2 == 0) {

System.out.println(number + " is an even number.");

} else {

System.out.println(number + " is an odd number.");

}

scanner.close();

}

}

**Explanation:**

1. Import the Scanner class to read input from the user.
2. Create a **main** method, which serves as the entry point of the program.
3. Create a Scanner object named **scanner** to read input from the user.
4. Prompt the user to enter an integer by printing "Enter an integer: ".
5. Read the integer entered by the user and store it in the **number** variable.
6. Use an **if** statement to check if the number is even or odd. The condition **number % 2 == 0** checks if the number is divisible by 2 (i.e., it's even).
7. If the condition is true, print a message stating that the number is even; otherwise, print a message stating that it's odd.
8. Close the Scanner object to release resources.

**Exercise 02:**

Write a Java program that calculates the grade for a student based on their exam score. The program should take the exam score as input and then determine and display the corresponding grade according to the following grading scale:

Score >= 90: A

Score >= 80: B

Score >= 70: C

Score >= 60: D

Score < 60: F

Solution:

import java.util.Scanner;

public class GradeCalculator {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter the exam score: ");

int score = scanner.nextInt();

char grade;

if (score >= 90) {

grade = 'A';

} else if (score >= 80) {

grade = 'B';

} else if (score >= 70) {

grade = 'C';

} else if (score >= 60) {

grade = 'D';

} else {

grade = 'F';

}

System.out.println("The grade is: " + grade);

scanner.close();

}

}

**Explanation:**

1. Import the Scanner class to read input from the user.
2. Create a **main** method as the entry point of the program.
3. Create a Scanner object named **scanner** to read input from the user.
4. Prompt the user to enter the exam score by printing "Enter the exam score: ".
5. Read the exam score entered by the user and store it in the **score** variable.
6. Declare a **grade** variable to store the calculated grade.
7. Use a series of **if-else if** statements to determine the grade based on the score, following the grading scale.
8. Print the calculated grade to the console.
9. Close the Scanner object to release resources.

**Exercise 03:**

Write a Java program that calculates the day of the week for a given date. The program should take three inputs: the day, month, and year, and then determine and display the day of the week for that date.

You can use Zeller's Congruence formula for this purpose:

h = (q + (13 \* (m + 1)) / 5 + K + (K / 4) + (J / 4) - 2 \* J) % 7

Where:

* h is the day of the week (0 = Saturday, 1 = Sunday, 2 = Monday, ..., 6 = Friday)
* q is the day of the month
* m is the month (3 = March, 4 = April, ..., 14 = February)
* J is the century (year / 100)
* K is the year within the century (year % 100)

Solution:

import java.util.Scanner;

public class DayOfWeekCalculator {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter the day (1-31): ");

int day = scanner.nextInt();

System.out.print("Enter the month (1-12): ");

int month = scanner.nextInt();

System.out.print("Enter the year: ");

int year = scanner.nextInt();

if (month < 3) {

month += 12;

year -= 1;

}

int J = year / 100;

int K = year % 100;

int h = (day + (13 \* (month + 1)) / 5.0 + K + (K / 4.0) + (J / 4.0) - 2 \* J) % 7;

String[] daysOfWeek = {"Saturday", "Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday"};

System.out.println("The day of the week for the given date is: " + daysOfWeek[h]);

scanner.close();

}

}

**Explanation:**

1. Import the Scanner class to read input from the user.
2. Create a **main** method as the entry point of the program.
3. Create a Scanner object named **scanner** to read input from the user.
4. Prompt the user to enter the day, month, and year.
5. Read the day, month, and year entered by the user and store them in respective variables.
6. Adjust the month and year values if the month is January or February. In this case, we treat those months as months 13 and 14 of the previous year.
7. Calculate the values of J and K based on the provided year.
8. Use Zeller's Congruence formula to calculate the day of the week (h).
9. Create an array **daysOfWeek** to map the calculated value of **h** to the corresponding day of the week.
10. Print the calculated day of the week to the console.

**Exercise 04:**

Write a Java program that determines whether a given year is a leap year or not. The program should take a year as input and then determine and display whether it's a leap year or not. A leap year is a year that is divisible by 4, except for years that are divisible by 100 but not divisible by 400.

Solution:

import java.util.Scanner;

public class LeapYearChecker {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

int year = scanner.nextInt();

if ((year % 4 == 0 && year % 100 != 0) || (year % 400 == 0)) {

System.out.println(year + " is a leap year.");

} else {

System.out.println(year + " is not a leap year.");

}

scanner.close();

}

}

**Explanation:**

1. Import the Scanner class to read input from the user.
2. Create a **main** method as the entry point of the program.
3. Create a Scanner object named **scanner** to read input from the user.
4. Prompt the user to enter a year by printing "Enter a year: ".
5. Read the year entered by the user and store it in the **year** variable.
6. Use an **if** statement to check whether the year is a leap year or not. The condition **(year % 4 == 0 && year % 100 != 0) || (year % 400 == 0)** checks the leap year conditions:
   * It should be divisible by 4 (year % 4 == 0).
   * It should not be divisible by 100 (year % 100 != 0), except if it's also divisible by 400 (year % 400 == 0).
7. If the condition is true, print a message stating that the year is a leap year; otherwise, print a message stating that it's not a leap year.
8. Close the Scanner object to release resources.

**Exercise 05:**

Write a Java program that helps a user determine the type of a given triangle based on its three side lengths. The program should take the lengths of the three sides as input and then determine and display whether the triangle is equilateral, isosceles, or scalene.

Solution:

import java.util.Scanner;

public class TriangleClassifier {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter the length of side 1: ");

double side1 = scanner.nextDouble();

System.out.print("Enter the length of side 2: ");

double side2 = scanner.nextDouble();

System.out.print("Enter the length of side 3: ");

double side3 = scanner.nextDouble();

if (side1 == side2 && side2 == side3) {

System.out.println("This is an equilateral triangle.");

} else if (side1 == side2 || side2 == side3 || side1 == side3) {

System.out.println("This is an isosceles triangle.");

} else {

System.out.println("This is a scalene triangle.");

}

scanner.close();

}

}

**Explanation:**

1. Import the Scanner class to read input from the user.
2. Create a **main** method as the entry point of the program.
3. Create a Scanner object named **scanner** to read input from the user.
4. Prompt the user to enter the lengths of three sides of the triangle.
5. Read the lengths of the sides entered by the user and store them in **side1**, **side2**, and **side3** variables.
6. Use **if-else** statements to determine the type of triangle based on the side lengths:
   * If all three sides are equal, it's an equilateral triangle.
   * If at least two sides are equal, it's an isosceles triangle.
   * Otherwise, it's a scalene triangle.
7. Print the type of triangle to the console.
8. Close the Scanner object to release resources.

**Exercise 06:**

Write a Java program that helps a user determine whether a given year is a "special" year. A year is considered special if the sum of its digits is divisible by 3. The program should take a year as input and then determine and display whether it's a special year or not.

Solution:

import java.util.Scanner;

public class SpecialYearChecker {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

int year = scanner.nextInt();

int sumOfDigits = 0;

int tempYear = year;

// Calculate the sum of the digits of the year

while (tempYear != 0) {

sumOfDigits += tempYear % 10;

tempYear /= 10;

}

if (sumOfDigits % 3 == 0) {

System.out.println(year + " is a special year.");

} else {

System.out.println(year + " is not a special year.");

}

scanner.close();

}

}

**Explanation:**

1. Import the Scanner class to read input from the user.
2. Create a **main** method as the entry point of the program.
3. Create a Scanner object named **scanner** to read input from the user.
4. Prompt the user to enter a year by printing "Enter a year: ".
5. Read the year entered by the user and store it in the **year** variable.
6. Initialize **sumOfDigits** to 0. This variable will be used to calculate the sum of the digits of the year.
7. Create a temporary variable **tempYear** to hold a copy of the **year** value for digit extraction.
8. Use a **while** loop to calculate the sum of the digits of the year. In each iteration, extract the last digit of **tempYear**, add it to **sumOfDigits**, and remove that digit from **tempYear** by dividing it by 10.
9. Use an **if** statement to check if the sum of the digits (**sumOfDigits**) is divisible by 3.
10. If the condition is true, print a message stating that the year is special; otherwise, print a message stating that it's not special.
11. Close the Scanner object to release resources.

**Exercise 07:**

Write a Java program that helps a user determine the season of the year based on the month and day. The program should take the month and day as input and then determine and display the corresponding season (spring, summer, fall, or winter).

import java.util.Scanner;

public class SeasonFinder {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter the month (1-12): ");

int month = scanner.nextInt();

System.out.print("Enter the day (1-31): ");

int day = scanner.nextInt();

String season;

if (month >= 3 && month <= 5) {

season = "spring";

} else if (month >= 6 && month <= 8) {

season = "summer";

} else if (month >= 9 && month <= 11) {

season = "fall";

} else {

season = "winter";

}

System.out.println("The season is " + season);

scanner.close();

}

}

**Explanation:**

1. Import the Scanner class to read input from the user.
2. Create a **main** method as the entry point of the program.
3. Create a Scanner object named **scanner** to read input from the user.
4. Prompt the user to enter the month (1-12) and the day (1-31).
5. Read the month and day entered by the user and store them in the **month** and **day** variables, respectively.
6. Declare a **season** variable to store the calculated season.
7. Use **if-else** statements to determine the season based on the month:
   * If the month is between March (3) and May (5), it's spring.
   * If the month is between June (6) and August (8), it's summer.
   * If the month is between September (9) and November (11), it's fall.
   * Otherwise, it's winter.
8. Print the calculated season to the console.
9. Close the Scanner object to release resources.

**Exercise** **08**:

Write a Java program that calculates the cost of a coffee order based on the size and type of coffee chosen by the user. The program should take the size (small, medium, or large) and type (regular or decaf) of coffee as input and then determine and display the total cost of the coffee order.

Assume the following prices:

* Small regular coffee: $2.00
* Small decaf coffee: $2.50
* Medium regular coffee: $2.50
* Medium decaf coffee: $3.00
* Large regular coffee: $3.00
* Large decaf coffee: $3.50

Solution:

import java.util.Scanner;

public class CoffeeOrderCalculator {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.println("Coffee Sizes: Small, Medium, Large");

System.out.print("Enter the coffee size: ");

String size = scanner.nextLine().toLowerCase();

System.out.println("Coffee Types: Regular, Decaf");

System.out.print("Enter the coffee type: ");

String type = scanner.nextLine().toLowerCase();

double price = 0.0;

switch (size) {

case "small":

price += (type.equals("regular")) ? 2.00 : 2.50;

break;

case "medium":

price += (type.equals("regular")) ? 2.50 : 3.00;

break;

case "large":

price += (type.equals("regular")) ? 3.00 : 3.50;

break;

default:

System.out.println("Invalid coffee size.");

scanner.close();

return;

}

System.out.println("Total cost: $" + price);

scanner.close();

}

}

Explanation:

1. Import the Scanner class to read input from the user.
2. Create a **main** method as the entry point of the program.
3. Create a Scanner object named **scanner** to read input from the user.
4. Display the available coffee sizes (small, medium, large) and coffee types (regular, decaf).
5. Prompt the user to enter the coffee size and coffee type.
6. Read the user's input for size and type, convert them to lowercase for case-insensitive comparison, and store them in the **size** and **type** variables.
7. Initialize the **price** variable to 0.0 to keep track of the total cost.
8. Use a **switch** statement to calculate the price based on the selected size and type, considering the provided prices.
9. Display the total cost of the coffee order.
10. Close the Scanner object to release resources.

**Exercise 09:**

Write a Java program that helps a user determine whether a given year is a leap year or not, and also calculate the number of days in February for that year. The program should take a year as input and then determine and display whether it's a leap year or not, and how many days February has in that year.

A leap year is a year that is divisible by 4, except for years that are divisible by 100 but not divisible by 400.

Solution:

import java.util.Scanner;

public class LeapYearAndFebruaryDays {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

int year = scanner.nextInt();

boolean isLeapYear = (year % 4 == 0 && year % 100 != 0) || (year % 400 == 0);

int februaryDays = isLeapYear ? 29 : 28;

if (isLeapYear) {

System.out.println(year + " is a leap year.");

} else {

System.out.println(year + " is not a leap year.");

}

System.out.println("February has " + februaryDays + " days in " + year + ".");

scanner.close();

}

}

**Explanation:**

1. Import the Scanner class to read input from the user.
2. Create a **main** method as the entry point of the program.
3. Create a Scanner object named **scanner** to read input from the user.
4. Prompt the user to enter a year by printing "Enter a year: ".
5. Read the year entered by the user and store it in the **year** variable.
6. Use a boolean variable **isLeapYear** to calculate whether the year is a leap year based on the provided leap year conditions.
7. Use the **isLeapYear** value to determine the number of days in February (either 28 or 29 days).
8. Print whether the year is a leap year or not based on the **isLeapYear** value.
9. Print the number of days in February for the given year.

**Exercise 10**

**Question:** Write a Java program that calculates the Body Mass Index (BMI) for a user based on their weight in kilograms and height in meters. The program should take the weight and height as input, calculate the BMI, and then determine and display the BMI category based on the calculated value:

* BMI < 18.5: Underweight
* 18.5 ≤ BMI < 24.9: Normal weight
* 25 ≤ BMI < 29.9: Overweight
* BMI ≥ 30: Obese

**Solution:**

import java.util.Scanner;

public class BMICalculator {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

double weight = scanner.nextDouble();

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

double height = scanner.nextDouble();

double bmi = weight / (height \* height);

System.out.printf("Your BMI is %.2f\n", bmi);

if (bmi < 18.5) {

System.out.println("You are underweight.");

} else if (bmi >= 18.5 && bmi < 24.9) {

System.out.println("You have a normal weight.");

} else if (bmi >= 25 && bmi < 29.9) {

System.out.println("You are overweight.");

} else {

System.out.println("You are obese.");

}

scanner.close();

}

}

**Explanation:**

1. Import the Scanner class to read input from the user.
2. Create a main method as the entry point of the program.
3. Create a Scanner object named scanner to read input from the user.
4. Prompt the user to enter their weight in kilograms.
5. Read the weight entered by the user and store it in the weight variable.
6. Prompt the user to enter their height in meters.
7. Read the height entered by the user and store it in the height variable.
8. Calculate the BMI using the formula: bmi = weight / (height \* height).
9. Print the calculated BMI to the console, formatted to two decimal places.
10. Use a series of if-else statements to determine the BMI category based on the calculated BMI value.
11. Print the BMI category to the console.
12. Close the Scanner object to release resources.

**Exercise 11**

**Question:** Write a Java program that checks whether a given string is a palindrome. The program should take a string as input, remove any spaces, convert it to lowercase, and then determine and display whether the string is a palindrome.

**Solution:**

import java.util.Scanner;

public class PalindromeChecker {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

String input = scanner.nextLine();

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

String reversedInput = new StringBuilder(cleanedInput).reverse().toString();

if (cleanedInput.equals(reversedInput)) {

System.out.println("The string is a palindrome.");

} else {

System.out.println("The string is not a palindrome.");

}

scanner.close();

}

}

**Explanation:**

1. Import the Scanner class to read input from the user.
2. Create a main method as the entry point of the program.
3. Create a Scanner object named scanner to read input from the user.
4. Prompt the user to enter a string.
5. Read the string entered by the user and store it in the input variable.
6. Remove any spaces from the input string and convert it to lowercase using replaceAll and toLowerCase methods.
7. Reverse the cleaned input string using the StringBuilder class.
8. Compare the cleaned input string with its reversed version to check if they are equal.
9. Print a message indicating whether the string is a palindrome or not.
10. Close the Scanner object to release resources.

**Exercise 12**

**Question:** Write a Java program that generates a random number between 1 and 100 and then prompts the user to guess the number. The program should give feedback on whether the guessed number is too low, too high, or correct. The process should repeat until the user guesses the correct number.

**Solution:**

import java.util.Random;

import java.util.Scanner;

public class NumberGuessingGame {

public static void main(String[] args) {

Random random = new Random();

Scanner scanner = new Scanner(System.in);

int randomNumber = random.nextInt(100) + 1;

int guess = 0;

System.out.println("Guess a number between 1 and 100:");

while (guess != randomNumber) {

guess = scanner.nextInt();

if (guess < randomNumber) {

System.out.println("Too low. Try again.");

} else if (guess > randomNumber) {

System.out.println("Too high. Try again.");

} else {

System.out.println("Correct! The number was " + randomNumber);

}

}

scanner.close();

}

}

**Explanation:**

1. Import the Random and Scanner classes.
2. Create a main method as the entry point of the program.
3. Create a Random object named random to generate random numbers.
4. Create a Scanner object named scanner to read input from the user.
5. Generate a random number between 1 and 100 using random.nextInt(100) + 1 and store it in the randomNumber variable.
6. Initialize the guess variable to 0.
7. Print a message prompting the user to guess a number between 1 and 100.
8. Use a while loop to keep prompting the user for guesses until they guess the correct number.
9. Inside the loop, read the user's guess and store it in the guess variable.
10. Use if-else statements to compare the guessed number with the random number and provide feedback:
    * If the guess is too low, print "Too low. Try again."
    * If the guess is too high, print "Too high. Try again."
    * If the guess is correct, print "Correct! The number was " followed by the random number.
11. Close the Scanner object to release resources.

**Exercise 13**

**Question:** Write a Java program that reads a list of integers from the user, stores them in an array, and then calculates and displays the sum, average, maximum, and minimum of the numbers in the array.

**Solution:**

import java.util.Scanner;

public class ArrayStatistics {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter the number of integers: ");

int n = scanner.nextInt();

int[] numbers = new int[n];

System.out.println("Enter " + n + " integers:");

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

numbers[i] = scanner.nextInt();

}

int sum = 0;

int max = numbers[0];

int min = numbers[0];

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

sum += numbers[i];

if (numbers[i] > max) {

max = numbers[i];

}

if (numbers[i] < min) {

min = numbers[i];

}

}

double average = (double) sum / n;

System.out.println("Sum: " + sum);

System.out.println("Average: " + average);

System.out.println("Maximum: " + max);

System.out.println("Minimum: " + min);

scanner.close();

}

}

**Explanation:**

1. Import the Scanner class to read input from the user.
2. Create a main method as the entry point of the program.
3. Create a Scanner object named scanner to read input from the user.
4. Prompt the user to enter the number of integers and store the value in the n variable.
5. Create an array named numbers to store the integers.
6. Prompt the user to enter n integers and store them in the array.
7. Initialize the variables sum, max, and min. Set max and min to the first element of the array.
8. Use a for loop to iterate through the array:
   * Calculate the sum of the numbers.
   * Update max if the current number is greater than the current max.
   * Update min if the current number is less than the current min.
9. Calculate the average by dividing the sum by n.
10. Print the sum, average, maximum, and minimum to the console.
11. Close the Scanner object to release resources.

**Exercise 14**

**Question:** Write a Java program that helps a user determine whether a given year is a leap year or not, and also calculate the number of days in February for that year. The program should take a year as input and then determine and display whether it's a leap year or not, and how many days February has in that year.

**Solution:**

import java.util.Scanner;

public class LeapYearAndFebruaryDays {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

int year = scanner.nextInt();

boolean isLeapYear = (year % 4 == 0 && year % 100 != 0) || (year % 400 == 0);

int daysInFebruary = isLeapYear ? 29 : 28;

if (isLeapYear) {

System.out.println(year + " is a leap year.");

} else {

System.out.println(year + " is not a leap year.");

}

System.out.println("February has " + daysInFebruary + " days in " + year + ".");

scanner.close();

}

}

**Explanation:**

1. Import the Scanner class to read input from the user.
2. Create a main method as the entry point of the program.
3. Create a Scanner object named scanner to read input from the user.
4. Prompt the user to enter a year and store the value in the year variable.
5. Determine whether the year is a leap year using the condition: (year % 4 == 0 && year % 100 != 0) || (year % 400 == 0).
6. Based on whether the year is a leap year or not, set daysInFebruary to 29 or 28.
7. Print whether the year is a leap year and the number of days in February.
8. Close the Scanner object to release resources.