# Lab: Methods

Test your tasks in the Judge system: <https://judge.softuni.org/Contests/4417>

## Sign of Integer Number

Write a program that:

* Reads an **integer number** **N** from the console
* Create a method that prints the **sign** of an entered number **N**:
* Print**:**
* If the number is positive **(N > 0)**: "**The number {number} is positive.**"
* If the number is negative **(N < 0)**: "**The number {number} is negative.**"
* If the number is zero (**N = 0)**: "**The number {number} is zero.**"

answer:

public class SignOfNumber

{

// Method to print the sign of a number

public static void PrintSign(int N)

{

if (N > 0)

{

Console.WriteLine($"The number {N} is positive.");

}

else if (N < 0)

{

Console.WriteLine($"The number {N} is negative.");

}

else

{

Console.WriteLine($"The number {N} is zero.");

}

}

public static void Main(string[] args)

{

int number = Convert.ToInt32(Console.ReadLine());

// Call the method to print the sign of the entered number

PrintSign(number);

}

}

### **Example**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2 | The number 2 is positive. |

|  |  |
| --- | --- |
| **Input** | **Output** |
| -5 | The number -5 is negative. |

|  |  |
| --- | --- |
| **Input** | **Output** |
| 0 | The number 0 is zero. |

## Grades

Write a program that:

* Reads a **grade** (floating-point number) from the console
* Create a **method** that receives a **grade** between **2.00** and **6.00**
* Prints the corresponding **grade in words**
* "**Fail**" - If the grade is in range **2.00 – 2.99** (inclusively)
* "**Average**" - If the grade is in range **3.00 – 3.49** (inclusively)
* "**Good**" - If the grade is in range **3.50 – 4.49** (inclusively)
* "**Very** **good**" - If the grade is in range **4.50 – 5.49** (inclusively)
* "**Excellent**" - If the grade is in range **5.50 – 6.00** (inclusively)

answer:

public class GradeToWords

{

// Method to print the grade in words based on specified ranges

public static void PrintGrade(double grade)

{

if (grade >= 2.00 && grade <= 2.99)

{

Console.WriteLine("Fail");

}

else if (grade >= 3.00 && grade <= 3.49)

{

Console.WriteLine("Average");

}

else if (grade >= 3.50 && grade <= 4.49)

{

Console.WriteLine("Good");

}

else if (grade >= 4.50 && grade <= 5.49)

{

Console.WriteLine("Very good");

}

else if (grade >= 5.50 && grade <= 6.00)

{

Console.WriteLine("Excellent");

}

else

{

Console.WriteLine("Invalid grade input. Grade must be between 2.00 and 6.00.");

}

}

public static void Main(string[] args)

{

double grade;

// Read input from console and parse to double

if (double.TryParse(Console.ReadLine(), out grade))

{

// Call the method to print the grade in words

PrintGrade(grade);

}

else

{

Console.WriteLine("Invalid input. Please enter a valid floating-point number.");

}

}

}

### **Example**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Input** | **Output** |  | **Input** | **Output** |  | **Input** | **Output** |
| 3.33 | Average | 4.50 | Very good | 2.99 | Fail |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Input** | **Output** |  | **Input** | **Output** |  | **Input** | **Output** |
| 5.70 | Excellent | 3.70 | Good | 2.00 | Fail |

## Printing Triangle

Write a program that:

* Reads an **integer number N** from the console
* Create a method for printing triangle depending on value of the number N

using System;

public class TrianglePrinter

{

// Method to print the desired triangle pattern

public static void PrintTriangle(int N)

{

// Print the top part of the triangle

for (int i = 1; i <= N; i++)

{

// Print numbers from 1 to i

for (int j = 1; j <= i; j++)

{

Console.Write(j + " ");

}

Console.WriteLine();

}

// Print the bottom part of the triangle

for (int i = N - 1; i >= 1; i--)

{

// Print numbers from 1 to i

for (int j = 1; j <= i; j++)

{

Console.Write(j + " ");

}

Console.WriteLine();

}

}

public static void Main(string[] args)

{

int number;

// Read input from console and parse to integer

if (int.TryParse(Console.ReadLine(), out number))

{

// Call the method to print the triangle pattern based on N

PrintTriangle(number);

}

else

{

Console.WriteLine("Invalid input. Please enter a valid integer.");

}

}

}

### **Example**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Input** | **Output** |  | **Input** | **Output** |  | **Input** | **Output** |
| 3 | 1  1 2  1 2 3  1 2  1 | 4 | 1  1 2  1 2 3  1 2 3 4  1 2 3  1 2  1 | 2 | 1  1 2  1 |

## Calculate Rectangle Area

Write a program that:

* Reads **two integer** numbers from the console**: width** and **length**
* Create a method which returns **rectangle area** with given **width** and **length**

**Hint:** Rectangle area can be calculated when you multiply width and length of the rectangle.

using System;

public class RectangleAreaCalculator

{

// Method to calculate and return the area of a rectangle

public static int CalculateRectangleArea(int width, int length)

{

return width \* length;

}

public static void Main(string[] args)

{

int width;

// Read width input from console and parse to integer

if (int.TryParse(Console.ReadLine(), out width))

{

int length;

// Read length input from console and parse to integer

if (int.TryParse(Console.ReadLine(), out length))

{

// Call the method to calculate the area of the rectangle

int area = CalculateRectangleArea(width, length);

Console.WriteLine($"{area}");

}

else

{

Console.WriteLine("Invalid input for length. Please enter a valid integer.");

}

}

else

{

Console.WriteLine("Invalid input for width. Please enter a valid integer.");

}

}

}

### **Example**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Input** | **Output** |  | **Input** | **Output** |  | **Input** | **Output** |
| 3  4 | 12 | 6  8 | 48 | 5  10 | 50 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Input** | **Output** |  | **Input** | **Output** |  | **Input** | **Output** |
| 8  7 | 56 | 3  7 | 21 | 2  8 | 16 |

## Repeat String

Write a program that:

* Reads a **text (string)** and **repeat** **count (integer number)** from the console
* Write a method that receives a string and a repeat count
* The method should return a new string, containing the initial one, repeated **count** times without space

using System;

public class StringRepeater

{

// Method to repeat a string without spaces based on repeat count

public static string RepeatString(string text, int repeatCount)

{

// Check if repeatCount is positive

if (repeatCount <= 0)

{

throw new ArgumentException("Repeat count must be a positive integer.");

}

// Create a new string builder to construct the repeated string

System.Text.StringBuilder sb = new System.Text.StringBuilder();

// Append the text to the string builder repeatCount times

for (int i = 0; i < repeatCount; i++)

{

sb.Append(text);

}

// Return the final repeated string

return sb.ToString();

}

public static void Main(string[] args)

{

string text = Console.ReadLine();

int repeatCount;

// Read repeat count input from console and parse to integer

if (int.TryParse(Console.ReadLine(), out repeatCount))

{

try

{

// Call the method to repeat the string

string repeatedString = RepeatString(text, repeatCount);

Console.WriteLine($"{repeatedString}");

}

catch (ArgumentException ex)

{

Console.WriteLine(ex.Message);

}

}

else

{

Console.WriteLine("Invalid input for repeat count. Please enter a valid integer.");

}

}

}

### **Example**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Input** | **Output** |  | **Input** | **Output** |  | **Input** | **Output** |
| abc  3 | abcabcabc | String  2 | StringString | Re  3 | ReReRe |

## Math Power

Write a program that:

* Reads **two integer numbers** from the console: **base numbe**r and **power**
* Create a method, which receives two numbers as parameters:
* The first number – the **base**
* The second number – the **power**
* The method should return the **base** raised to the given **power**

**using System;**

public class PowerCalculator

{

// Method to calculate base raised to the power

public static double CalculatePower(int baseNumber, int exponent)

{

// Using Math.Pow to calculate baseNumber^exponent

return Math.Pow(baseNumber, exponent);

}

public static void Main(string[] args)

{

int baseNumber;

// Read base number input from console and parse to integer

if (int.TryParse(Console.ReadLine(), out baseNumber))

{

int power;

// Read power input from console and parse to integer

if (int.TryParse(Console.ReadLine(), out power))

{

// Call the method to calculate the power

double result = CalculatePower(baseNumber, power);

Console.WriteLine($"{result}");

}

else

{

Console.WriteLine("Invalid input for power. Please enter a valid integer.");

}

}

else

{

Console.WriteLine("Invalid input for base number. Please enter a valid integer.");

}

}

}

### **Example**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Input** | **Output** |  | **Input** | **Output** |  | **Input** | **Output** |
| 3  4 | 81 | 2  8 | 256 | 4  2 | 16 |

## Greater of Two Values

Write a program that:

* Reads a **type (string)** and **two values** of this type from the console
* Entered type can be one of the following values: "**int**", "**char**" or "**string**"
* Create methods **which can compare int, char or string**
* Return the **biggest of the two values**

**using System;**

public class ValueComparer

{

// Method to compare two integers and return the biggest one

public static int CompareIntegers(int a, int b)

{

return Math.Max(a, b);

}

// Method to compare two characters and return the biggest one

public static char CompareCharacters(char a, char b)

{

return a > b ? a : b;

}

// Method to compare two strings and return the one that is lexicographically largest

public static string CompareStrings(string a, string b)

{

return string.Compare(a, b) > 0 ? a : b;

}

public static void Main(string[] args)

{

string type = Console.ReadLine()?.ToLower(); // Read type and convert to lowercase for case-insensitivity

switch (type)

{

case "int":

if (int.TryParse(Console.ReadLine(), out int int1))

{

if (int.TryParse(Console.ReadLine(), out int int2))

{

int biggestInt = CompareIntegers(int1, int2);

Console.WriteLine($"{biggestInt}");

}

else

{

Console.WriteLine("Invalid input for the second integer. Please enter a valid integer.");

}

}

else

{

Console.WriteLine("Invalid input for the first integer. Please enter a valid integer.");

}

break;

case "char":

Console.WriteLine("Enter the first character:");

char char1 = Console.ReadKey().KeyChar; // Read character directly from console

Console.WriteLine(); // Move to the next line after reading the character

Console.WriteLine("Enter the second character:");

char char2 = Console.ReadKey().KeyChar; // Read character directly from console

Console.WriteLine(); // Move to the next line after reading the character

char biggestChar = CompareCharacters(char1, char2);

Console.WriteLine($"{biggestChar}");

break;

case "string":

string str1 = Console.ReadLine();

string str2 = Console.ReadLine();

string biggestString = CompareStrings(str1, str2);

Console.WriteLine($"{biggestString}");

break;

default:

Console.WriteLine("Invalid type entered. Please enter 'int', 'char', or 'string'.");

break;

}

}

}

### **Example**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Input** | **Output** |  | **Input** | **Output** |  | **Input** | **Output** |
| int  2  16 | 16 | char  a  z | z | string  aaa  bbb | bbb |

## Multiply Evens by Odds

Write a program that **multiplies the sum** of **all even digits** of a number **by the sum of all odd digits** of the same number:

* Read **an integer number** from the console
* Create a method called **GetMultipleOfEvenAndOdds()**
* Create a method **GetSumOfEvenDigits()**
* Create **GetSumOfOddDigits()**
* You may need to use **Math.Abs()** for negative numbers

using System;

public class DigitMultiplier

{

// Method to calculate the sum of even digits of a number

public static int GetSumOfEvenDigits(int number)

{

number = Math.Abs(number); // Ensure positive value for calculation

int sum = 0;

while (number > 0)

{

int digit = number % 10;

if (digit % 2 == 0)

{

sum += digit;

}

number /= 10;

}

return sum;

}

// Method to calculate the sum of odd digits of a number

public static int GetSumOfOddDigits(int number)

{

number = Math.Abs(number); // Ensure positive value for calculation

int sum = 0;

while (number > 0)

{

int digit = number % 10;

if (digit % 2 != 0)

{

sum += digit;

}

number /= 10;

}

return sum;

}

// Method to multiply the sum of even digits by the sum of odd digits

public static int GetMultipleOfEvenAndOdds(int number)

{

int sumEven = GetSumOfEvenDigits(number);

int sumOdd = GetSumOfOddDigits(number);

return sumEven \* sumOdd;

}

public static void Main(string[] args)

{

if (int.TryParse(Console.ReadLine(), out int number))

{

int result = GetMultipleOfEvenAndOdds(number);

Console.WriteLine($"{result}");

}

else

{

Console.WriteLine("Invalid input. Please enter a valid integer.");

}

}

}

### **Example**

|  |  |  |
| --- | --- | --- |
| **Input** | **Output** | **Comment** |
| -12345 | 54 | Evens: 2 4  Odds: 1 3 5  Even sum: 6  Odd sum: 9  6 \* 9 = 54 |

|  |  |  |
| --- | --- | --- |
| **Input** | **Output** | **Comment** |
| 3453466 | 220 | Evens: 4 4 6 6  Odds: 3 5 3  Even sum: 20  Odd sum: 11  20 \* 11 = 220 |

## Orders

Write a program that:

* Reads a **string** on the first line from the console, representing a **product** - "**coffee**", "**water**", "**coke**" or "**snacks**"
* Reads an **integer** on the second line from the console, representing the **quantity** of the product
* Create a method that calculates and prints the total price of an order
* The method should receive two parameters: **product** and **quantity**
* The prices for a single item of each product are:
* **coffee – 1.50**
* **water – 1.00**
* **coke – 1.40**
* **snacks – 2.00**
* Print the result, **formatted to the second digit**

**using System;**

public class OrderCalculator

{

// Method to calculate and print the total price of an order

public static void CalculateTotalPrice(string product, int quantity)

{

// Define prices per item using a dictionary

var prices = new Dictionary<string, double>

{

{ "coffee", 1.50 },

{ "water", 1.00 },

{ "coke", 1.40 },

{ "snacks", 2.00 }

};

// Check if the product exists in the dictionary

if (prices.ContainsKey(product))

{

double pricePerItem = prices[product];

double totalPrice = pricePerItem \* quantity;

// Print the result formatted to two decimal places

Console.WriteLine($"{totalPrice:F2}");

}

else

{

Console.WriteLine("Invalid product. Please enter one of: coffee, water, coke, snacks");

}

}

public static void Main(string[] args)

{

// Read product and quantity from console input

string product = Console.ReadLine()?.ToLower(); // Convert to lowercase for case-insensitive comparison

if (int.TryParse(Console.ReadLine(), out int quantity) && quantity > 0)

{

// Call method to calculate and print total price

CalculateTotalPrice(product, quantity);

}

else

{

Console.WriteLine("Invalid quantity. Please enter a valid positive integer.");

}

}

}

### **Example**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Input** | **Output** |  | **Input** | **Output** |  | **Input** | **Output** |
| water  5 | 5.00 | coffee  2 | 3.00 | snacks  6 | 12.00 |