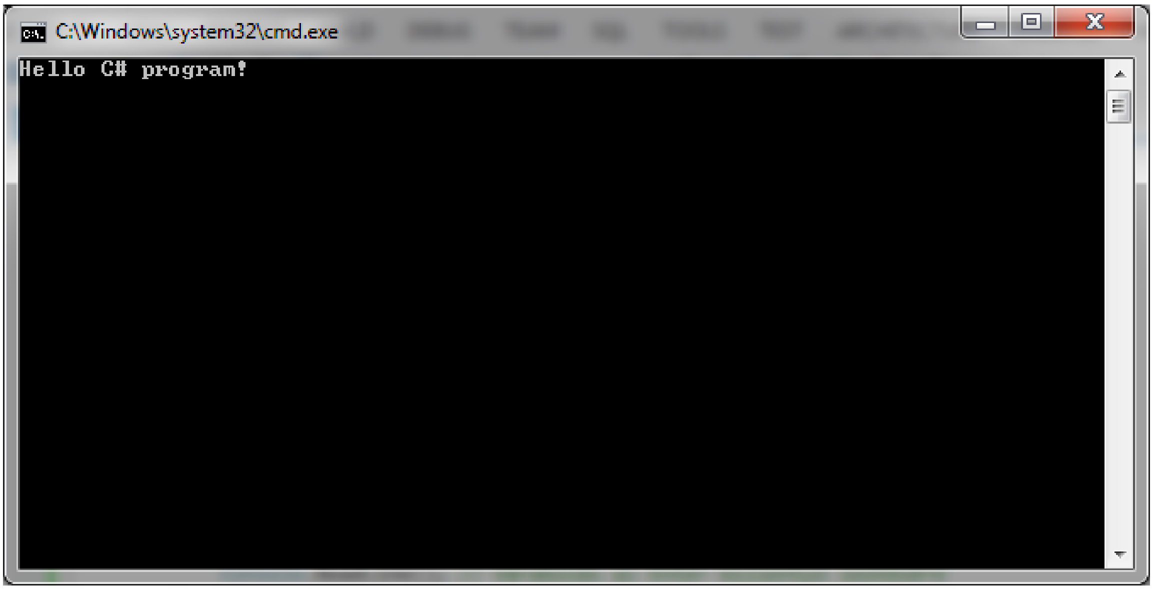
C# 1. Practice – Create simple projects

If the exercises cannot be completed by the end of the practice, they are required to be solved at home.

Exercises:

1. Create a console application that display the text „Hello C # program!” to the screen.
After you created the code use the *Build* menu in Visual Studio to translate your program and run it with the *Start* menu.



```
Hello C# program!
```

The program is waiting for the Enter key.

2. Write a program that reads a positive integer value, and consider whether even or odd and it is divisible by 5! The program will handle the case even if the user has entered the wrong data, for example a negative number. Although the task requests to examine the divisible by 5, please take care not to burn in the fixed 5 value.
   Test your program in different data! Test you program with text input as well. To improve the program use the int. TryParse() method!

3. Create a program that reads an integer number and can evaluate that the number is positive, negative, or zero. Display the result in text format!

4. Create a program that reads an essay scores, and sets the mark pursuant to section limits! The program should display the scores and the assigned mark value. (For example: The essay score is 68, the mark is 3.)
   Make sure that the score limits not to burn permanently in the program!

Score limits	Marks
0 – 50	1
51 – 65	2
66 – 75	3
76 – 85	4
86 – 100	5

Loops

A **loop** is a structure that allows repeated execution of a block of statements. Within a looping structure, a Boolean expression is evaluated. If it is true, a block of statements called the **loop body** executes, and the Boolean expression is evaluated again. As long as the expression is true, the statements in the loop body continue to execute and the loop-controlling Boolean expression continues to be reevaluated. When the Boolean evaluation is false, the loop ends. (While, do..while loops)
You can use a **while loop** to execute a body of statements continuously as long as the loop's test condition continues to be true. The **do loop** checks the loop-controlling Boolean expression at the bottom of the loop after one repetition has occurred.
A loop for which the number of iterations is predetermined is called a **definite loop** or **counted loop**. This is the **fo**r loop.
The **foreach** loop will use later.

5. Create a program

a) to compute the factorial of a number (n!)! A factorial of a number is the product of all numbers between the given number and 1. For example: n! = 1 * 2 * 3 * ... * n
b) to compute the factorial of the first n numbers!
c) Write method to compute the factorial of a number, and call it 2-3 different data.

6. Write a C# program that displays prime numbers belongs to a given interval! The endpoints of the interval is given by the user in a controlled manner. To determine whether a number is prime or not, write a method. ((`static bool PrimeNum(int number){})`
   To determine if a number is a prime number, try to divide the number by all integers between 2 and square root of the number. (The square root of a number: Math.Sqrt(number)).

7. Write a console-based application that allows the user to enter any number of integer values continuously (in any order) until the user enters 999. Display the sum and the average of the values entered, not including 999.

8. Create a program to solve the next problem!
   Create a structure called Employee that includes the employee's name, the number of hours worked by that employee and the hourly rate. The structure contains a method that can calculate the wages to be paid and another method, which returns a text. The text includes the employee's name and the calculated wage.
   The Main method reads the data of these types of workers from the console. After reading all data of one employee the program displays the name and the wage (salary) of the employee, then ask if there are any more employee's data to be processed. If there is no more data to be processed, calculate and display the total amount of wages paid to employees.