

## Exercise 1

Create a new Console Application and inside it create two new variables to your program that stores string values. One of them is going to store your first name and the other one your last name, so give them informative names.

Then let the program output: "Hello <firstname> <lastname>! I'm glad to inform you that you are the test subject of my very first assignment!"

### Subjects Covered:

Variables

## Exercise 2

Create new Console Application, create three new variables of type DateTime and let them store yesterdays, todays and tomorrows date. Remember to give them informative names.

Then let your program output them to the screen like:

"Todays date is <todays date>"

"Tomorrows date is <tomorrows date>"

"Yesterdays date was <yesterdays date>"

Information about DateTime can be found at:

<http://www.dotnetperls.com/datetime>

### Subjects Covered:

DateTime

## Exercise 3

Create a new Console Application and write the code inside that prompts the user to input his first and last name. Also create two new variables that stores string values and save the input from the user into these (So that one variable stores the first name and the other stores the last name). Then output the full name to the screen on one line.

### Subjects Covered:

String variables

## Exercise 4

Create a new Console Application and inside it create two new string variables. The first one will contain the original text of "The quick fox Jumped Over the DOG".

Use string manipulation methods (IndexOf, Remove, Replace, Insert) to change the string into the following text "The brown fox jumped over the lazy dog", where all characters matches exactly and save the change text in the second string variable.

Code: `String originalText = "The quick fox Jumped Over the DOG";`

### Subjects Covered:

String manipulation

## Exercise 5

Below is a given string. Use string manipulations to get the [1,2,3,4,5] part from the string and store it into a new string variable. Then remove the values 2 and 3 and insert 6-10 into it in the end, comma separated so that the result is [1,4,5,6,7,8,9,10] and let your program output that to the console window.

Code: `String str = "Arrays are very common in programming, they look something like: [1,2,3,4,5]"; Console.ReadLine();`

### Subjects Covered:

String manipulation

## Exercise 6

Write the program code that lets the user input 2 integers from the console. Then let the program output the biggest, smallest, difference ( - ), sum ( + ), product ( \* ) and ratio ( / ) between the two numbers.

### Subjects Covered:

Mathematical operators

## Exercise 7

Write the program code that lets the user input the radius (as a double). Then calculate the area and volume of a circle respective sphere with the given radius. (area =  $2 \pi r^2$ , volume =  $(4 \times \pi \times r^3)/3$ ). Output the result to the screen.

### Subjects Covered:

Mathematical operators

## Exercise 8

Let the user input a decimal number. Then output square root of the number and the number raised to the power of 2 and 10. That is  $\sqrt{n}$ ,  $n^2$  (square root 2),  $n^{10}$  (square root 10).

### Subjects Covered:

Mathematical operations

## Exercise 10

Write the program code that will ask to user to choose between 3 different options using a switchstatement.

1. Let the first option invoke a method that let the user input two numbers (a and b). Check that b is not equal to zero and then output a divided by b to the screen. If b is equal to zero, display an error message to the user.
2. Let the second option invoke the method used in exercise 4 (The quick fox jumped over the lazy dog).
3. Let the third option toggle the foreground colour between two different colours each time you choose it. (Use an if-statement to check the current colour)

### Subjects Covered:

Switch-statement

## Exercise 11

In the exercise method, write the code that lets the user input an integer number. Check that the given number is greater than 0. If not display an error message. Then write two for-loops, where one start from zero and counts up to the given number (0,1,2,3,4..., n) and another for-loop that starts counting from the given number down to zero (n,...,3,2,1,0). In both loops, check each number if it's evenly divisible by 2. If it is, change the console colour to red, else to green.

### Subjects Covered:

Iterations

## Exercise 12

Write a program that output the multiplication table for 1 to 10 using nested for-loops. The format is not of great important but to make the code readably, add a tab after each number using the escape character `\t`.  
1 2 3 4  
5 6 7 8 9 10 2 4 6 8 10 12 14 16 18 20 3 6 9 12 ....

### Subjects Covered:

Nested for-loops

## Exercise 13

Write a program that first generates a random number between 1 and 500 and stores it into a variable (see the Random class). Then let the user make a guess for which number it is. If the user types the correct number, he should be presented with a message (including the number of guesses he has made). If he types a number that is greater or smaller than the given number, display either "Your guess was too small" or "Your guess was too big". The program should keep executing until the user input the correct guess.

### Subjects Covered:

Random

## Exercise 14

Write a program that keeps asking the user for input numbers, until he enters -1. Store the amount of numbers the user have entered and the sum of the numbers added together. When the user types -1, the program should display the sum and the average of the numbers.

Write a program that keeps asking the user for input numbers, until he enters -1. Store the amount of numbers the user have entered and the sum of the numbers added together. When the user types -1, the program should display the sum and the average of the numbers.

-----

Enter a number: 33  
Enter a number: 74  
Enter a number: 123  
Enter a number: -1  
Sum: 230  
Average: 76,666666666667

### **Subjects Covered:**

Iteration

## **Exercise 15**

Write the program code that asks the user for a number. Then display all numbers that the number is divisible by. Example entering 12, should output 6, 4, 3, 2 and 1. Tip: use modulo and a loop. • Write the program code that outputs the 3 first perfect numbers. A perfect number is a number where all its positive divisors sums up to the actual number. The first number is 6, where  $3 + 2 + 1 = 6$  and the second is 28, where  $14 + 7 + 4 + 2 + 1 = 28$ . Tip: look at the previous exercise and build on top of it.

### **Subjects Covered:**

Iteration

## **Exercise 16**

Write a program that asks the user for a number. Use this number to output the Fibonacci series up until that number. Entering 10 should then output: 0, 1, 1, 2, 3, 5, 8, 13, 21 and 34

### **Subjects Covered:**

Mathematical functions

## Exercise 17

Let the user input a string, then check if the string is a palindrome sentence. A palindrome is a word or sentence that reads the same in both directions. Example of palindrome sentences are: Loops at a spool, wet stew and level. However, the spaces might look different depending on which direction you read it, so these should be excluded in your calculations, and a tip is to use some string manipulation to remove them.

-----  
Enter a word to check:  
level "level" is a palindrome  
Enter a word to check: test  
"test" is not a palindrome  
Enter a word to check: loops at a spool  
"loops at a spool" is a palindrome

### Subjects Covered:

String manipulation

## Exercise 18

Create a new empty integer array of 10 elements. Loop through the array and assign each element to a new random value. Create a new empty array of doubles, having the same size as the previous array. Loop through that array and assign the values to 1 divided by the value on the same position of the previous array. So if the first array has the value 42 on position 3, the second array should hold the double value  $1 / 42$ . Finally, loop through both arrays and output the values to the screen using a foreach-loop.

### Subjects Covered:

Array, iteration

## Exercise 19

Create a program that outputs a price that the customer (user) needs to pay. This should be an integer value. Then let the user input the sum he hands the cashier. Let your program then calculate the change that the customer should get back in different coin unit. For example, if the user hands the cashier 500 kr. and the price is 376 kr., the change will be 124. This can be divided up into 100x1 kr. + 20x1 kr. + 4x1 kr. The goal here is to get as few coins as possible.

Tip: Use an array to store the different coin units, like 100, 50, 20 etc. When you have calculated the change, go through the coin units from largest to smallest and calculate how many of each type the customer should get back. Here is a perfect scenario when integer division and modulus is viable to use.

### Subjects Covered:

Arrays, Modulus

## Exercise 20

Create two arrays with arbitrary size and fill one with random numbers. Then copy over the numbers from the array with random numbers so that the even numbers are located in the rear (the right side) part of the array and the odd numbers are located in the front part (the left side).

### Subjects Covered:

Arrays, Random



## Exercise 21

Let the user input a string with numbers comma separated like "1,2,34,83,19,45". Create the code to separate the numbers in the string into an array and find the min, max and average value. Print these out to the screen. Tip: use the Split-function on the String-object. Keep in mind that the Split-method returns an array of Strings, so you have to convert each value to an integer before you can do the calculations.

### **Subjects Covered:**

Split, Array manipulations

## Exercise 22

The company See sharp AB have discovered that the users on their forums use a very harsh language when interacting with each other. So now they have asked you to write a swear word filter to censor these occurring words. Before implementing this filter on their website, they want a demonstration in form of a console program.

The program should ask the user for a textual input. Then it should check for all occurrences of swear words. Store the swear words in an array and check the textual input against the array and use string manipulation to replace all swear words with something more appropriate.

### **Subjects Covered:**

Array manipulation and comparison

## Exercise 23

Create a program that generates 7 unique numbers into an array. The numbers should be between 1 and 40. Each of the numbers may only appear once in the array! The numbers should be generated using the Random-class, and should be different each time you run the program.

### **Subjects Covered:**

Random

## Exercise 24

Create a program that can shuffle a deck of cards. The cards can be represented as an array of integers, like [1,1,1,1,2,2,2,2,3,3,...n]. Then make it possible to draw 1 card from the deck and add to another array. (The card should then disappear from the card deck and appear in the array with the drawn cards. Tip: You can use `Array.Resize( ref array, newSize)` to change the size of an existing array, and `Array.Copy`. There is however not a requirement that the array needs to be in a specific order after a card has been drawn. Complete the functions below.

```
-----  
static int DrawCard(ref int[] deck)  
{  
    // code to draw a card here  
}  
-----  
static void ShuffleCards(ref int[] deck)  
{  
    // code to shuffle the deck here  
}
```

### Subjects Covered:

Array manipulation

## Exercise 25

Create a separate function (from the exercise method) that asks the user to input a valid integer value. The function should keep executing until the user has inputted a valid integer value. Use a Try-catch combined with a while-loop. If the user inputs a none-valid number, display an error asking him/her to try again. Then in the exercise method, call the method you just wrote twice to retrieve 2 integers from the user and store them in variables. Then divide one of the number with the other and use a try-catch to catch any potential division by zero. Display the result to the screen.

Note: Use the correct Exception type, example `FormatException`, `DivideByZeroException` and `ArgumentNullException` and multiple catch-statements to display a friendly error message depending on which error occurs.

### Subjects Covered:

Errors, try-catch

## Exercise 26

In this exercise, you are going to print out the folder path to some different locations on your computer. These can be found in the `Environment`-class. To print one of the common folders on your computer, you can use the `Environment.GetFolderPath`, passing it one of the values from the `Environment.SpecialFolder` enumeration. Print out the following locations on your computer:

Example: `Environment.GetFolderPath(Environment.SpecialFolder.Desktop)`

- My documents
- My Pictures
- The folder containing information about cookies
- Current Directory (found inside in the `Environment` class)

In the same method, create a new file on the desktop, using `File.Create(filepath)`, by retrieving the folder path to the desktop and append the filename at the end of the file path. When creating a new file using `File.Create`, a new `FileStream` will be created, and this must be closed using the `Close` method.

Note: Depending on which computer you are running your application, some special folders might not exist.

### **Subjects Covered:**

Diretories, Enviroments

## **Exercise 27**

Inside your project in Visual Studio, right-click on your project and add a new text file. Open the file and add at least 10 names on one row each. Right click on the file and hit Properties. In the Properties window, make sure that Copy to Output Directory is set to Copy Always. This will make sure that the files are copied to the binfolder when your program is compiled (see figure 5).

Now, use a `StreamReader` object (found inside `System.IO`) to read the names from the file and output them to the console window.

Note: There are a few possibilities when reading from a file, but one thing to be careful about, is that file might have a larger size than the available memory on your computer. Using `ReadToEnd()`, which reads every byte in the file would then cause a `OutOfMemoryException`. This is however not an issue with this exercise.

### **Subjects Covered:**

Reading a file

