

Exercises about data types

Solve them in Visual Studio or use <https://dotnetfiddle.net/>.

Exercise 04.01: Adding doubles

- Create a variable (amount) and give it the value 123,4567.
 - Do the exercise with data type “double”.
- Add 234,5678 to variable amount.
- Show the result to the console.



Do the exercise again with the data type “decimal”.

Exercise 04.02: Properties of a Date

- Define a variable that contains the date and time of the current moment. (DateTime.Now)
- Show on the console in different lines:
 - Year: the year of now.
 - Month: the month of now.
 - Day: the day of now.
 - Hour: the hour of now.
 - Minutes: the minutes of now.
 - Seconds: the seconds of now.

Notes

COPY PASTE)

Exercise 04.03: A lot of decimals

- Define a variable of the data type double and add the value 0,1234567890123456789 to it.
- Show on the console that value.



What do you see?

You have done already this kind of exercise, so this is a kind of repeating yourself.

Do you know how to correct this?

Exercise 04.04: Password generator 01

The code below generates random a number between 1 and 20.

- Lower number is included.
- Bigger number is excluded.

```
static void Main()
{
    Random bytRandom = new Random();

    Console.WriteLine(bytRandom.Next(1, 20));
}
```

- Create using this method “Next” to generate a password with minimum 8 and maximum 14 random characters.
- Get a random number.
- Find in the alphabet “abcdefghijklmnopqrstuvwxyz” the corresponding character.
 - Pay attention that there is sometimes a “z” in the password.

An example of a correct result

adqrazzgup

Notes

COPY PASTE)

Exercise 04.05: Password generator 02

- Build on the result of 04.04 or restart.
- Every letter can be used only once. This means that a letter can't occur twice in the password.

An example of a correct result

adqrzgupb

Exercise 04.06: Password generator 03

- Build on the result of 04.05 or restart.
- Capitals, digits and &§!+/.? can also be used.
- When your solution method is correct, the routine / method that generated the password should not change at all.

Exercise 04.07: Keep track of things



You will have unexpected results in this routine. Mark them with a comment in your exercise.

This exercise is easy, but hard because of the many steps. Add code, test, add something, test again, and so on ...

- Declare a variable with name “myByte” of data type “byte”.
- Give that variable the value 123.
- Show the value of “myByte” to the console.
- Declare a variable with name “myShort” of data type “short”.
- Give that variable the value of “myByte”.
- Show the value of “myShort” to the console.
- Declare a variable with name “myInt” of data type “int”.

Notes

.....
.....
.....
.....
.....
.....
.....

COPY PASTE)

Documentation & Information

- Give that variable the value of “myShort”.
- Show the value of “myInt” to the console.
- Declare a variable with name “myLong” of data type “long”.
- Give that variable the value of “myInt”.
- Show the value of “myLong” to the console.
- Declare a variable with name “myFloat” of data type “float”.
- Give that variable the value of “myLong”.
- Show the value of “myFloat” to the console.
- Declare a variable with name “myDouble” of data type “double”.
- Give that variable the value of “myFloat”.
- Show the value of “myDouble” to the console.
- Declare a variable with name “myDecimal” of data type “decimal”.
- Give that variable the value of “myDouble”.
- Show the value of “myDecimal” to the console.
- Give the value 123456 to “myLong”.
- Give the value of “myLong” towards “myInt”. (Use casting)
- Show the value of “myLong” to the console.
- Give the value 123456789123456789 to “myLong”.
- Give the value of “myLong” towards “myInt”. (Use casting)
- Show the value of “myInt” to the console.
- Give the value 1234 to “myInt”.
- Give the value of “myInt” towards “myShort”. (Use casting)
- Show the value of “myShort” to the console.
- Give the value 1234567 to “myInt”.
- Give the value of “myInt” towards “myShort”. (Use casting)
- Show the value of “myShort” to the console.
- Give the value 123 to “myInt”.
- Give the value of “myInt” towards “myByte”. (Use casting)
- Show the value of “myByte” to the console.
- Give the value 1234 to “myInt”.
- Give the value of “myInt” towards “myByte”. (Use casting)
- Show the value of “myByte” to the console.

Notes

COPY PASTE)

Exercise 04.08: The Transport Problem (Brute Force)



This is not an optimal solution, but is it a great exercise to program loops (while, for, foreach, ...) and decisions (if, switch, ...)

You are a transport organiser in a company that makes refrigerators. There are 2 locations (factories) where refrigerators are assembled.

You have 3 shops where the refrigerators will be delivered.

Tomorrow, a total of 25 refrigerators must be delivered to the 3 shops.

- The first shop needs 10.
- The second shop needs 8.
- The third shop needs 7.

In the first factory there will be 11 refrigerators ready for transportation and in the second factory there will be 14 ready.

The cost of transportation from the factory towards the shops is a fixed amount per refrigerator.

- From the first factory to the first shop: € 8,00.
- From the first factory to the second shop: € 6,00.
- From the first factory to the third shop: € 10,00.
- From the second factory to the first shop: € 9,00.
- From the second factory to the second shop: € 5,00.
- From the second factory to the third shop: € 7,00.

Your task, calculate all possible combinations (this is brute force) and show on the console the cheapest possibility in delivery costs. And while you are busy, what is the solution that is the most expensive?

Notes

COPY PASTE)