Assignment – 1

Q1. Parse the string variables from below into the correct datatypes:

```
public static void Main(string[] args)
{
  string stringForFloat = "0.85"; // datatype should be float
  string stringForInt = "12345"; // datatype should be int
}
```

Q2. Declare a string variable and don't assign any value to it. Print on the console "Please enter your name and press enter". You can then enter your name or any other valid string like "John". Assign that entered string to the string variable which you have declared initially. The program should write on the console that string in Uppercase in the first line, then the same string in Lowercase in the second line. In the third line, it writes on the console the string with no trailing or preceding white space like if string entered as " John Cena " it should be written on the console as "JohnCena". And in the last line, it should write the Substring of the entered string on the console.

Q3. Let's create another console application for more practice.

This application asks the user to input a string in the first line like "Enter a string here: ".

In the Second Line, it should ask for the character to search in the string which you have entered in the first line like "Enter the character to search: "

On the third line, it should write the index of the first occurrence of the searched character from the string.

Now on concatenation...

It should then ask to enter the first name and once the name is written and the enter button is pressed, it should ask to enter the last name.

It should then show your full name printed in a single line like in my case the output will be "Denis Panjuta". Output can be different in your case. Try to store the full name in a variable, before displaying it.

Q4. Create a variable for each of the primitive datatypes. Leave the Object datatype out. And also please initialize each variable with a working value.
Then create two values of type string. The first one should say "I control text"

The second one should be a whole number. Then use the Parse method in order to convert that string to an integer. Add each an output for each of the variables and write it onto the console.

Q5. create three variables with names of your friends.
Then create a Method "GreetFriend" which writes something like: "Hi Frank, my friend!" onto the console, once it is called.
Where "Frank" should be replaced with the Name behind the argument given to the Method when it's called. So the method will need a parameter (decide which DataType will be best).
Greet all your three friends.

Q6. Write two new methods:

LowUpper method will take a string as input, create lower- and uppercase copies, and return the result of their join.

Example:

"lol" string will produse "lolLOL".

The Count method also will take a string, but it will not return. It only will print "The amount of characters is X." with X replaced with the actual amount of characters in the string on another line.

Example:

"lol" string will print "The amount of characters is 3.".

Alert!

Both of them should be static and public. So add "static" and "public" keywords like those written below for the Run method.

The result of execution for the default string should be:

hey there !HEY THERE !

The amount of characters is 22.

Q7. Implement method Check that will take an int as an argument and check if it is an odd number or even. Depending on the testing result, it should print a message on the new line "Odd" or "Even" respectively.

Q8. This time you have to complete the NestedCheck method. It is supposed to print the right message in the terminal depending on the given value:

- when the value is divisible by 3. You have to print "Divisible by 3.";
- when it is not divisible by 3, but divisible by 7 print "Divisible by 7."
- Otherwise:
    - when it is an odd number, print "Odd number."
    - if not, print "Even number."

Example:

the number is equal to 3, "Divisible by 3." is printed;

the number is equal to 28, "Divisible by 7." is printed;

the number is equal to 1, "Odd number." is printed;

In the attachment, you can find the file OurSolution.txt. Before opening it, try to solve the exercise yourself.

Q9. Create a user Login System, where the user can first register and then Login in. The Program should check if the user has entered the correct username and password, wenn login in (so the same ones that he used when registering).

As we haven't covered storing data yet, just create the program in a way, that registering and logging in, happen in the same execution of it.

Q10. Create a user Login System, where the user can first register and then Login in. The Program should check if the user has entered the correct username and password, wenn login in (so the same ones that he used when registering).
As we haven't covered storing data yet, just create the program in a way, that registering and logging in, happen in the same execution of it.

Q11. Create an application with a score, highscore and a highscorePlayer.
Create a method which has two parameters, one for the score and one for the playerName.
When ever that method is called, it should be checked if the score of the player is higher than the highscore, if so, "New highscore is + " score" and in another line "New highscore holder is " + playerName - should be written onto the console, if not "The old highscore of " + highscore + " could not be broken and is still held by " + highscorePlayer.
Consider which variables are required globally and which ones locally.

Q12. Let's create a small application that takes a temperature value as input and checks if the input is an integer or not.If the input value is not an integer value, it should print to the console "Not a valid Temperature". And if the input value is the valid integer then it should work as mentioned below.

If input temperature value is <=15 it should write "it is too cold here" to the console.

If input temperature value is >= 16 and is <=28 it should write "it is ok" to the console.

If the input temperature value is > 28 it should write "it is hot here" to the console.

Make sure to use ternary operators and not if statements to check for the three conditions, however you can use if statement for the other conditions like to check if the entered value is a valid integer or not.

Q13. There are two methods you have to create (one is ForLoop and the other is WhileLoop):

In the ForLoop method you have to print on a new line each value from -3 to 3 included using the for loop;

In the WhileLoop method you need to print values from 3 to -3 using the while loop.

Q14. Implement conditions for the given while loop.

```
using System;

namespace Coding.Exercise

{

   public class Exercise

   {

      public static void Run()

      {

         int i = -10;

         while(true){

            // TODO

            if (i == 11)

            {

               Console.WriteLine(" FINAL BREAK REACHED! This should not happen!");

               break;

            }

            Console.WriteLine(i++);

         }

      }

   }

}
```

To pass the tests, your loop should skip all divisible by 3 values and stops running when i = 10.

*Warning: You cannot touch the given parts of the code! You can add your conditions only inside the loop!

*Warning2: This while loop is an infinite loop! To avoid it causing you issues we implemented the current if statement that you can find. This, with the proper solution in place, should not be needed anymore. Before running the tests, find a way to stop it without reaching the pre-placed break!

Q15. Imagine you are a developer and get a job in which you need to create a program for a teacher. He needs a program written in c# that calculates the average score of his students. So he wants to be able to enter each score individually and then get the final average score once he enters -1.
So the tool should check if the entry is a number and should add that to the sum. Finally once he is done entering scores, the program should write onto the console what the average score is.
The numbers entered should only be between 0 and 20. Make sure the program doesn't crash if the teacher enters an incorrect value.
Test your program thoroughly.

Q16. The main of this exercise is to create 3 constructors for the class Phone:

```
using System;

namespace Coding.Exercise
{
    public class Phone
    {
        public string Company;
        public string Model;
        public string ReleaseDay;
        // Place for your constructors
        public void Introduce()
        {
            Console.WriteLine("It is {0} created by {1}. It was released {2}.", Model, Company, ReleaseDay);
```

```
    }
  }
}
```

default constructor that sets all values to "unknown";

constructor that takes 2 arguments (company and model) and sets them respectively. The ReleaseDay should be set to "unknown";

constructor with 3 arguments (company, model, and release day);

Warning: You should not touch the given parts of the code! Create only 3 constructors.

Q17. In this exercise, you will practice creating properties in a class and validating the data before it is set. Specifically, you'll create a Book class with properties for the title and number of pages.

1. Within the Book class, create two private fields: a string called _title and an integer called _pages.
2. Implement properties for both _title and _pages:

For the Title property, ensure that it cannot be set to an empty string. If an empty string is passed, set it to "Unknown".

For the Pages property, ensure that it cannot be set to a value less than 1. If a value less than 1 is passed, set it to 1.

*Note: Remember to use getters and setters for creating properties.

*Note: Utilize conditional statements in the setter to perform validation.