Kristianstad University Sweden
Kristianstad University
SE-291 88 Kristianstad
Sweden
+46 44 20 30 00
www.hkr.se

Homework Module 1: C# Basics

(Solve the task on this page as a warm up before starting with the actual homework on the next page. You don't need to include it as part of your submission.)

A method of designing software is called "**Test-driven development**"; it means to write test program before the actual program is written. This is an example:
Assume your team works with a program that contains calculation operations. Your task is to write a piece of program that multiplies an input float number with 2.2, 3.5 and 4.75.
The code is simple:

```
float result1 = 0.00F, result2 = 0.00F, result3 = 0.00F;
float inputValue;
result1 = inputValue * 2.20F;
result2 = inputValue * 3.50F;
result3 = inputValue * 4.75F;
```

Important notice: **Please pay attention to '.' and ',' in floats if you are working in English or Swedish.**
You write a program that calculate the three results and display the result as table with three columns. If the input number is 45,58 the output looks like the table below (border lines not part of the program):

2,2	3,5	4,75
100,276	159,53	216,505

The test code is:

```
static void Main(string[] args)
{
    //Declare and initialize three float variables
    float result1 = 0, result2 = 0, result3 = 0;
    //Reads the input value
    string str = Console.ReadLine();
    /*Convert the string representation of the number
    to its single floating-point number equivalent */
    float inputValue = float.Parse(str);
    //Calculations
    result1 = inputValue * (float)2.20; //instead of (float) you could
    also write 2.20F
    result2 = inputValue * (float)3.50;
    result3 = inputValue * (float)4.75;
    //Display the result to verify the calculations. Note that the
    \t indicates escape sequence(tab)and \n indicates new line.
     Console.WriteLine("The rate are : 2,20\t 3,50\t 4,75" +
    "\nThe output is: {0}\t{1}\t{2}", result1, result2, result3);
}
}
```

The output is:

```
45,58
The rate are : 2,20      3,50    4,75
The output is: 100,276  159,53  216,505
```

With this test program you verify the syntax and the correctness of your code logic. Test the correctness of your program to see if it works for negative numbers in the input as well.

Kristianstad University
Sweden
Kristianstad University
SE-291 88 Kristianstad
Sweden
+46 44 20 30 00
www.hkr.se

Homework:

In this assignment, you shall create **only one** project. All the tasks shall be implemented in the same project (e.g. Homework1) as different methods (code blocks) named Task1(), Task2(), … Task8(), as it is shown below. Call all the Task-methods from the project's main method. Like this:

```
public class Program
    {
        public static void Main(string[] args)
        {
            Task1();
            Task2();
            ...
        }

        public static void Task1()
        {
          //Here you will write the code for Task1
        }

        public static void Task2()
        {
          //Here you will write the code for Task2

        }
        ...

    }
```

(Press **Ctrl + F5** to run your program)

You will learn about methods later in the course but for now all you need to do is to create the above format in your program and write the corresponding code for each task inside it's block.

Tasks:

1. Read two strings from the user input (keyboard), concatenate them and print the result string on the screen.

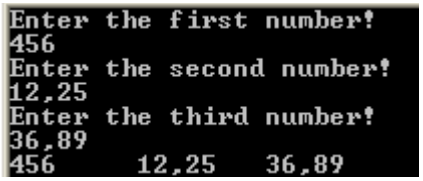
You can see an example of the output below:

Your program will ask the user to enter the first string, then it will ask the user for the next string and finally the two strings will be concatenated and the result will be printed. For example, if the user enters for the first time: "The family name of John is", and for the second time: "Dale", the output may look like:

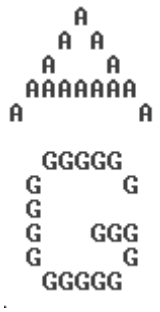
```
Please enter the first string:
The family name of John is
Please enter the second string:
Dale
The result is: The family name of John is Dale
Press RETURN to Quit.
```

2. Read an integer from the keyboard and determine and display whether it is odd or even. (Hint: Use the remainder operator)

3. Read three numbers from the keyboard and display it on one line on the screen as a table by using escape sequence \t. If the user enters "456", "12,25", "36,89", the output looks like:



4. Write your initials on the screen by using the first letters of your first and last name, such as:



5. Read two numbers from the keyboard and assign them to two variables of the type *int*, named x and y, compute the sum, difference, product, division and remainder of these two integers, then display the result on the screen. The pattern of the output could be like that: "The sum of *x[Value] and y[Value]* is […]"

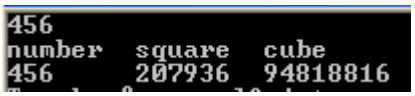
6. Read the radius r of a circle from the keyboard as integer and display the circle's **diameter**, **circumference** and **area** using π value. C# has defined a data type that represents floating-point numbers that contains decimal point, such as 3.14159. Use the following formulas, where r is radius:

Diameter = 2r
Circumference = $2\pi r$
Area = πr^2

You can define PI yourself using this approximate value, or better, you can use the constant from the Math library (Math.PI).

7. Read a four digit integer, such as 5893, from the keyboard and display the digits separated from one another by a tab each.
Hint: Use both integer division and modulus operator (%) to pick off each digit. E.g., if the user enters 4567, the output looks like:

8. Read an integer from the keyboard and display the square and the cube of the number. If the user enters 456, the output looks like:



Minimum requirements for grades:
Grade 3: 5 tasks solved with well structured code.
Grade 4: 7 tasks solved with well structured code.
Grade 5: all tasks solved with nice and well-structured code

Other measures that could affect the grade in a positive (+) or negative (-) way

a. Elegant solution [+] (note: "elegant" is not always the same as extremely compact code)
b. Unnecessarily complex solution [–]
c. Well formatted and well-structured code with descriptive variable names [+]
d. Unreadable and messy code [–]
e. Commented code [+]

Note (applicable to all Homework Assignments)
The style of your source code will be considered when setting the grade.

f. Write well formatted and well-structured code. Messy code with poor layout and indentation is hard to understand and more likely to contain errors.

It is good to write short comments before each block of code or where you think it is necessary.

g. Try to use descriptive and understandable variable names, e.g.

- Good example: `string familyName;`
- Bad example: `string fn;`

Remember to compress the whole project folder, name it as your first and last name (like firstname_lastname_homework1**) and upload it on Canvas.**

If you have questions, you can use the discussion forum or send me an email at: *dawit.mengistu@hkr.se*

Good luck!

(Credits to:
Alexandru Christian & Nazila)