Practice Assignments Introduction Windows Forms and C#

Quiz questions, practical assignments and

answers to quiz questions

**Version:** 1.0 **Last updated:** 15 november 2023

Contents

[1 Practical assignments: Windows Forms App 3](#_Toc147862658)

[1.1 Programming Assignment 1: Exploring visual controls 3](#_Toc147862659)

[1.1.1 Screenshots 3](#_Toc147862660)

[1.1.2 Case description 3](#_Toc147862661)

[1.2 Programming Assignment 2: Toggle colors 5](#_Toc147862662)

[1.2.1 Case description 5](#_Toc147862663)

[1.2.2 User interaction 5](#_Toc147862664)

[1.3 Programming Assignment 3: BMI calculator 5](#_Toc147862665)

[1.3.1 Case description 5](#_Toc147862666)

[1.3.2 User interaction 5](#_Toc147862667)

[1.3.3 Screenshots 6](#_Toc147862668)

[1.3.4 Additional features 6](#_Toc147862669)

[1.4 Programming Assignment 5: Truck management 6](#_Toc147862670)

[1.4.1 Case description 6](#_Toc147862671)

[1.4.2 User interaction 6](#_Toc147862672)

[1.4.3 Screenshots 7](#_Toc147862673)

# Practical assignments: Windows Forms App

All the assignment in this chapter should be done with Visual Studio and the project type Windows Forms App.

## Programming Assignment 1: Exploring visual controls

Difficulty: C:\Users\874156\Desktop\flatastic-icons-part-1-by-custom-icon-design\png\16x16\star-2_5.png

The assignment covers the following learning goals:

* Understand how to use Windows Forms Application.
* Understand how properties and events function for controls.

### Screenshots

### Case description

In this assignment, you learn how to build a Windows Forms Application in Visual Studio. You will learn how to edit properties of the controls and use their events.

1. First create a new Windows Forms Application project by choosing “File -> New -> Project”. In case you did not chose C# when you installed Visual Studio, it might be that Visual Studio starts with another programming language. In that case, chose in “Template” for “Visual C#”. At this point, there are many options for the kind of applications (or apps) that you might want to implement in C#. We select the Windows Forms Application.  
   Visual Studio suggest the default name “WindowsFormsApp1” for your app. Change this to a better name, something like “MyFirstWindowsApp”. Be aware that the name should be an identifier.  
   Chose a location (a folder on your hard drive) in which your first app should be stored. For this purpose there is a button with text “Browse” to indicate the location to store this project.  
   Remember where you store it: you might need it for later (step 9).  
   The Solution name (at the bottom of the window) is the same as Name. We keep it as it is.
2. The text on the title-bar of the window is "Form1". Select the window (Form1) by clicking once on it. Now the property-window displays the properties of the window. Change the property Text in "My first program".
3. Add a button on your window by selecting a button in the Toolbox and add it to the window. In the screenshot above, its BackColor is yellow. Give this button a nice color (be creative; what is your favorite color?). Change the text of the button in something like “Who am I?” or “Information about me”.
4. Clicking this button should result in displaying information about you on the screen. You will have to program it. There are 2 ways of choosing an event: one way is illustrated in the lesson (in the Property-window choose for events and then the click-event). Now use the other way. Microsoft has chosen a “most-likely-meant-event” for every control. For a button, it is the click-event.  
   Double-click the button. Visual Studio will direct you to the event-handler for button1\_click. Place the following text in body of this event handler:  
   MessageBox.Show("My name is <your\_name>");   
   Run your app and click the button.
5. The next control (the red one in the above figure) is a trackbar. Place a trackbar on your window and change its Minimum and Maximum in, for instance, 0 and 100. Change its property Value in 20. Run your program to see what happens. Now you can scroll with this trackbar, but nothing happens. Also, you can give the trackbar a nice BackColor.
6. Place a progress bar (in the above picture the lower green/gray rectangle). Also place a label on your window (in the picture it has the text "50"). Normally the font is very small. Can you give this label a bigger font?
7. We would like the progress bar and the text in the label to “follow the trackbar” (if the user scrolls with the trackbar, its value should be displayed in the label and be visual in the progress bar).  
   Make the progress bar’s property Minimum equal to 0 and its Maximum equal to 100 (the same Minimum and Maximum as your trackbar)  
   Doubleclick on the trackbar to let Visual Studio generate the “most-likely-meant-event”-handler:  
   private void trackBar1\_Scroll(object sender, EventArgs e)  
   {  
   }  
     
   Type the code  
    this.progressBar1.Value = this.trackBar1.Value;  
    this.label1.Text = this.trackBar1.Value.ToString();  
   in this trackBar1\_Scroll-method, run your program and see what happens.
8. There are several kind of buttons. You already practiced with a normal button. Next, we will explore radiobuttons. You see three of them on the right side of the window.  
   The most important property of a radiobutton is the property “Checked”. It’s value is a boolean (true or false). When the user selects a radionbutton, its property “Checked” becomes true and you see it visually on the screen (a dot in the circle before the text). But there is more: if you place some radiobuttons on the form, at every moment there cannot be 2 or more selected. If you click on a radiobutton, the clicked radiobutton will become selected and all other radiobuttons will automatically be deselected. Run your app to check it.  
   If you push the first radiobutton, the window's backcolor should become red. This radiobutton has already a nice text. Choose some nice texts for the two other radiobuttons.  
   By the way, so far we did not change the name of any visual control. Since we have only 1 button, there will be hardly any confusion what is meant by “button1”. Part of pragmatics is to choose an appropriate name. Now we have 3 radiobuttons, so please, don’t name them radiobutton1, radiobutton2, radiobutton3, as proposed by Visual Studio. Rename them to something like rbMakeMeRed, rbMakeMe??? and rbMakeMe???.  
   When doubleclicking on a radiobutton, Visual Studio generates the “most-likely-meant-event”-handler CheckedChanged. For the radiobutton rbMakeMeRed, add the code  
    this.BackColor = Color.Red;  
    to it. There are some pre-defined colors. If you type Color, followed by a dot, they will pop up.  
   Do the same for the other radiobuttons.
9. Now you are so proud of your first program that you cannot wait to send it to your family. Maybe you want your picture on the window: just do it (use a picturebox to hold your image. Most probably your image does not have the same size as the picturebox. Ask a fellow student or the teacher how to handle it).  
   After running your app, Visual studio creates a subfolder with the same name as your project. In it you find the subfolder “bin” and in that folder “debug”. There are some files. One of them is the executable (.exe). This executable runs on Windows, even if you do not have Visual Studio. Thus, now you can send this app to your family.

## Programming Assignment 2: Toggle colors

Difficulty: 

The assignment covers the following learning goals:

* Implementing a simple if-statement.

### Case description

Create a new application which has one button on the form. When clicking this button, the application should toggle between 3 colors, for example red, green and blue (feel free to choose any colors).

### User interaction

If the background-color of the window is red, then clicking the button results in a green background. Similarly, if the background-color of the window is green, then clicking the button results in a blue background and when it is blue it results in a red background.

## Programming Assignment 3: BMI calculator

Difficulty: 

The assignment covers the following learning goals:

* Practice with if-statements and operators

### Case description

Create an application with GUI for Body Mass Index (BMI) calculator.

This application should determine whether you are too skinny, have normal weight or are overweight based on the entered weight and height. The BMI is calculated as follows:

BMI = weight / ( height2), where weight is in kilograms and height is in meters.

The following table indicates the value of BMI:

|  |  |
| --- | --- |
| **Body mass index** | **explanation** |
| bmi < 18,5 | underweight |
| 18,5 <= bmi < 25,0 | normal weight |
| 25,0 <= bmi < 30,0 | overweight |
| bmi >= 30,0 | obese |

### User interaction

The user should be able to input a weight and a height (e.g. in some textboxes, each having a nice name). The user should then be able to click a button, which results in calculating the BMI, displaying it somewhere on the window and also inform the user about underweight, normal, overweight, or obese. For example, in case of obese, show the text “Your BMI is extremely high. Please contact your doctor, because your health is in danger!”.

### Graphical user interface, application Description automatically generatedScreenshots

### Additional features

By coding: this.lblExplanation.Text = "your bmi is: " + bmi.ToString();

you get a lot of decimals as in the above screenshot, because the BMI is converted to a string in the default way.

There are other ways of converting. One of them is that you want to have a fixed number of decimals. In that case, you have to hand the pattern "0.00" to the ToString()-method.

The code this.lblExplanation.Text = "your bmi is: " + bmi.ToString("0.00");

converts the BMI to a string, rounded to 2 decimals.

## Programming Assignment 5: Truck management

IDE: Visual Studio

Difficulty: C:\Users\874156\Desktop\flatastic-icons-part-1-by-custom-icon-design\png\16x16\star-4_5.png

The assignment covers the following learning goals:

* Use operators in your app.

### Case description

The owner of a truck company needs help with managing his business. He needs an application to calculate how many trucks are required to transport a certain order (the number of boxes that a supermarket ordered). The transport company delivers the ordered boxes to supermaerkets, but they need to be able to calculate how many trucks are needed.

The company has three types of trucks with different capacities (i.e. amount of pallets and the size of pallets). Make sure it is possible for the user to be able to specify per truck type: 1) how many pallets fit in the truck, 2) hoe many boxes fit on a pallet (see screenshot).

### User interaction

Start a new project and add some controls to it. We start with the rectangles that look like panels, but they are not. A panel decorates the GUI. If you want to do so, but you also want to have a text at the top of it, use a groupbox.   
A groupbox has the same functionality as a panel, but it has some text on top of it. So first put two groupboxes on the window and then some other controls.

The user should be able to input the number of boxes and which type of truck is going to be used. Based on this data, your application should show the following:

* How many full trucks
* When applicable:
  + How many full pallets are remaining for the non-full truck
  + How many boxes are on the non-full pallet

For example, based on the inputted data in the screenshot below:

* 2 Full trucks
* 1 Partially filled truck with:
  + 14 Full pallets
  + 1 Partially filled pallet with 15 boxes

Oops, in the screenshot we forgot to add the necessary controls tor perform the task and to show the result. Decide for yourself what controls are necessary to make it work.

### Screenshots

Graphical user interface, text, application

Description automatically generated