Basics in Programming

By
Adam Higherstein

Free ebook
v. 0.1

Basics in Programming

Language: C#

Free ebook, version 0.1

by
Adam Higherstein

Table of Contents

Basics in Programming

Introduction

What is programming?

We give instructions to the computer: set of instructions is a program.
Computer is mainly the processor that can understand machine code.
So our instructions are compiled to machine code so that it can be executed by the computer.

A program contains
storages, data structures
functions, activities, operations

Programming languages
There are several programming languages, also for different purposes.

Here are the most used languages:
Java
Used in workstation and enterprise applications AND Android phones
It is also an Object Oriented Programming language (OOP)

C and C++
C is used in Embedded programming, games and so on
Procedural Programming language (not OOP!!)
C++
It is also an Object Oriented Programming language (OOP)
C++ is used in Game programming for different kinds of applications and for operating systems

C#
We use C# in this book. It is developed by Microsoft, used in ASP.NET, workstation software, games etc.
It is also an Object Oriented Programming language (OOP)

PHP
Used for web programming

JavaScript
Used for web programming

HTML
Web page contents markup language

Python
Used for different kinds of applications

SQL
Used for database queries

ObjectiveC
Used for IOS plattform

Assembler
It is a symbolic machine language

Take a look at the lists of programming languages. Here is for example a list of most popular programming languages:
http://spectrum.ieee.org/computing/software/the-2016-top-programming-languages

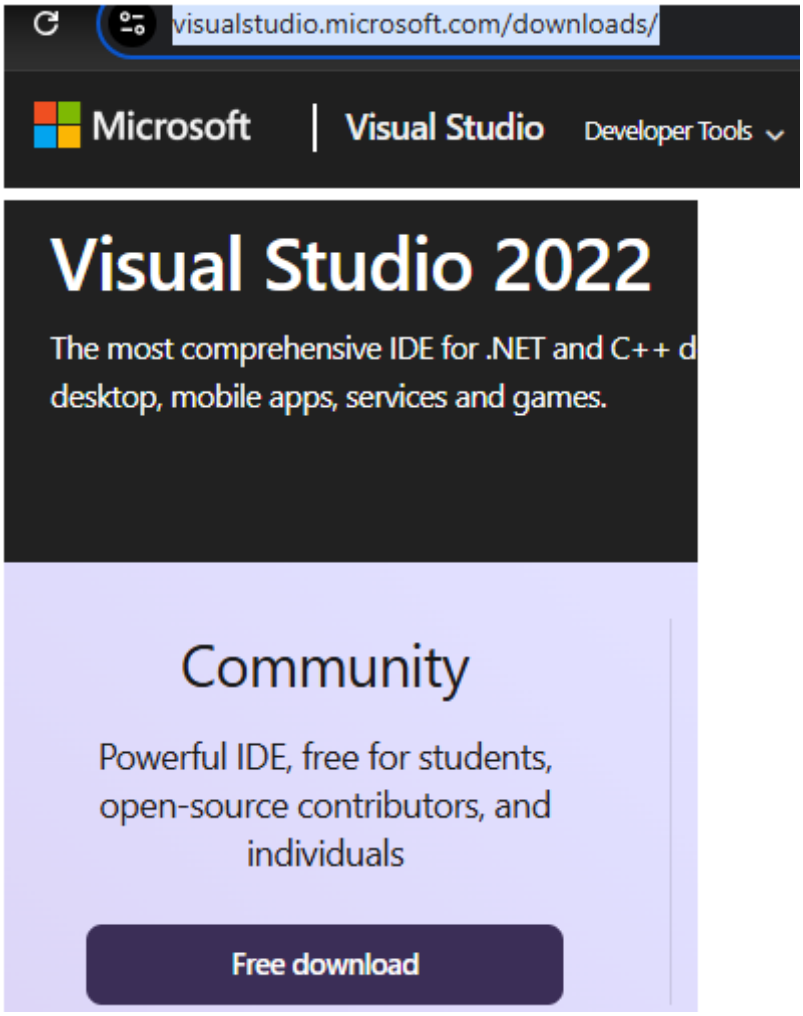
Our tool
In this book w use Visual Studio: it is a good tool for console programs and also for applications that have graphical user interface that we also handle in this book.

Visual studio is called also an IDE tool. IDE means "Integrated Development Environment". Same tool has many parts so that is is easier to create, debug and run the application...

Visual Studio is also a RAD (Rapid Application Development) tool: it means that is seems to be easy to

The concrete tool is Visual Studio Community version that can be downloaded for example from this site:
https://visualstudio.microsoft.com/downloads/

Here is a screen copy



Let’s get to know a bit about VS tool!

Task 1: install Visual Studio to your own computer!!

Here are good Microsoft’s instructions concerning installation and starting a new project

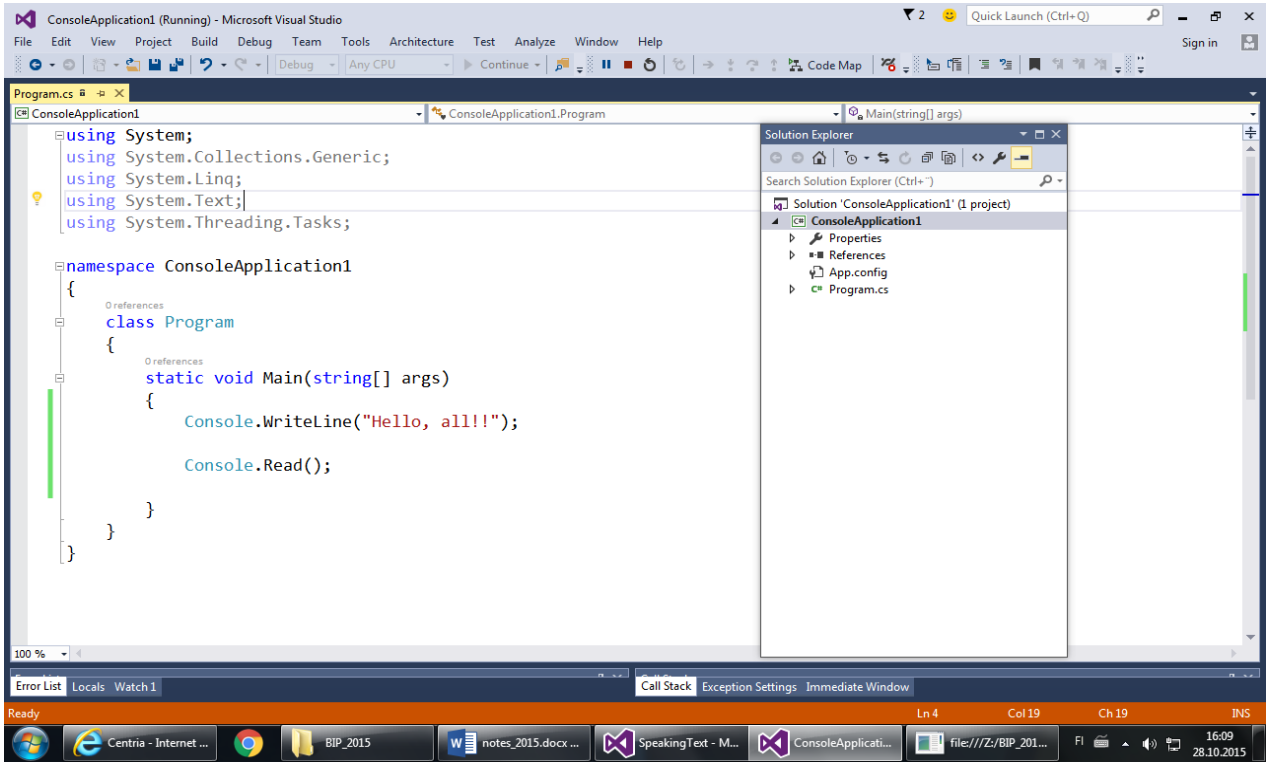
https://learn.microsoft.com/en-us/visualstudio/install/install-visual-studio?view=vs-2022

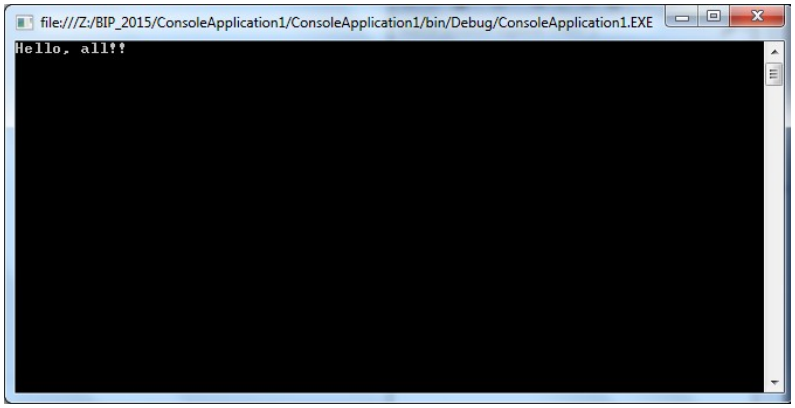
Visual Studio

Start a new console project, give it a name and go on.

Video: installing tool and starting a new project
Watch now this short video about starting a new project. We add there a greeting and run the program.
Parts of the tool window are explained, too.

Here below you see a new project in Visual Studio.





Variables

Variables are storages used by the program. Memory for variables is allocated from computer's memory. That memory is called RAM (Random Access Memory) memory.

Variables have to be defined before they can be used.

In definition we need to tell data type and name of the variable.

Data type defines what kinds of values we can store to a variable:
Are they integers, e.g. values like 1, 20, 10000
Are they floating point (decimal) values, e.g. values like 2.35, 100.5555
Are they Characters, e.g. values like 'a', '4'
Are they boolean values, e.g. values like true, false
Are they texts (strings) , e.g. values like "Kokkola", "USA"

Here below we have a list of different data types and how they are used.

C# Data types names

data type	explanation, range	example
int	Integer, whole number +- 2 000 000 000	int a = 1000000;
long	longer integer +- 9 000 000 000 000 000 000	long worldpopulation = 8000000000;
short	shorter integer +- 30000	short pietarsaariPopulation = 25000;
sbyte	allocates memory 1 byte, for shortest integers +- 127	sbyte daysInMonth = 30;
byte	allocates memory 1 byte, for shortest integers 0 – 255	byte daysInMonth = 30;
char	2 bytes, for Unicode characters	char character = 'A';
bool	Boolean value, true or false	bool hasPcAtHome = true;
float	4 bytes single precision +- 3*10^40 about 7 significant digits are stored	float weight = 50.25F;
double	8 bytes double precision +- 3*10^300 about 14 significant digits are stored default data type for decimal values	double density = 1.23456789;
decimal	16 bytes big precision range: 8 * 10^30	decimal value = 12345678.1234567M;
ushort	0 – 65000 unsigned short	
uint	0 – 4 billions unsigned int	
ulong	0 – 20 * 10^18 unsigned long	

String	for texts	String name = “Kokkola”;

Variables and data types: examples

```
// variables and datatypes
int populationOfFinland = 5500000;

int populationOfKokkola = 50000;

 // amount of days in a month: 1 - 31
byte daysInMonth = 30;

// population of world = 7500000000
long populationOfWorld = 7500000000;

// value like 123.345  (6 significant digits)
float value1 = 123.345F;
// note: floating point value is seen as a double value in C#!
// so, put there F to tell that I want see the value as float

// value like 123.3456789  (10 significant digits)
double value2 = 123.3456789;

// some letter as 'p'
char letter = 'p';

// some text as "Helsinki"
String text = "Helsinki";

// to tell if user has work or not  (boolean type)
bool hasWork = true;

// print values
Console.WriteLine("Variable value1 is " + value1);
Console.WriteLine("Variable value2 is " + value2);
Console.WriteLine("Variable letter is " + letter);
Console.WriteLine("Variable text is " + text);
Console.WriteLine("Variable hasWork is " + hasWork);

// stop console
Console.Read();
```

Video: creating and using variables

Now it is time to study arithmetic operations: may be we have different kinds on math calculations in our programs. We fortunately now already how to add or subtract values manually – now we have know how to use programming language to do same things.

Here below is a list of arithmetic operations with examples.

Arithmetic operators

Operator	Explanation	Example

+	Addition	int a = 5;
		int b = 3;
		int c = a + b;
		c is now 8

-	Subtraction	c = a – b; c is now 2
*	Multiplication	c = a * b; c is now 15
%	Remainder	c = a % b; c is now 2

/	Division	c = a/b;
		c is now 1!!!
		Note:
		integer division is in use in several programming languages:
		if both operands are whole numbers, the result is a whole number!
		To avoid this, convert one of operands to floating point type!
		AND:
		the floating point result has to be put to floating point variable!
		
		double c;
		c = (double) a/b;
		OR
		c = 1.0 * a/b;
		c is 1.6666666;

Math operations: examples

```
Task 1
Our program uses Ohm's law to calculate the resistance.
 User gives voltage and current.
```

Solution

```
double U, I, R;
U = 100;
I = 25;
R = U/I;

Console.WriteLine(“R is “ + R + “ ohms”);
```

Task 2

User gives the speed of the car (km/h) and the distance (km). Program calculates amount of time.

a) in hours

b) in whole hours and minutes

Solution

a)

```
double v;
double s;
double t;
v = 75;
s = 120;

v = s/t and then
t = s/v;

Console.WriteLine(“ t is “ + t + “  hours”);
```

b)

```
int wholeHours = (int) t;   //   1
int minutes = (int) ( (t – wholeHours) * 60);   // 0.6 * 60 = 36

Console.WriteLine(“whole hours: “ + wholeHours + “ and remaining minutes: “ + minutes);
```

Task 3

Our program calculates BMI.

Solution

```
double weight = 100;
double height = 200;

double heightInMeters = height/100
double bmi = weight/(heightInMeters * heightInMeters);

Console.WriteLine(“Your BMI is “ + bmi);
```

Task 4

Create a euro converter: dollars to euros.

Solution

```
double dollars = 1234;
double euros = dollars * 0.918015;

Console.WriteLine(“Dollars “ + dollars + “ is in euros “ + euros);
```

Task 5

Convert given seconds to hours, minutes, seconds.

Solution

```
int allSeconds = 123456;
int hours = allSeconds / 3600;  // hours is  34

allSeconds = allSeconds – hours * 3600;  // allSeconds is  1056
int minutes = allSeconds/60;  // minutes 17

int seconds = allSeconds – minutes *  60; // 36 seconds
```

Task 6

Convert euros to 5, 10, 20, 50, 100, 200, 500 euros bills.

Solution

```
int euros = 1234;
int b500 = euros/500;  // 2
euros = euros – b500*500; // 234
int b200 = euros/200;  // 1
euros = euros – b200*200; // 34
int b100 = euros/100;  // 0
euros = euros –b100*100; // 34
int b50 = euros/50;  // 0
euros = euros –b50 * 50; // 34
int b20 = euros/20; // 1
euros = euros –b20*20; // 14
int b10 = euros/10;  // 1
int b5 = euros –b10*10; // 0
int remainingEuros = euros – b5*5;  // 4
```

Some special math operators

Incrementing by 1: ++
Decrementing by 1: --

Example:

```
int a = 10;
a++;  // same as a = a + 1;
a--;  // same as a = a - 1;
```

AND
if ++ or -- is before the variable, operation is done first and then the statement
if ++ or -- is after the variable, operation is done after the statement

Examples:

```
// ++ --
            int a = 10;
            Console.WriteLine("a is " + a);
            a++;
            Console.WriteLine("a is " + a);
            a--;
            Console.WriteLine("a is " + a);

            Console.WriteLine("a is " + (++a) );

            Console.WriteLine("a is " + (a++));

            Console.WriteLine("a is " + a);
```

Other operators (assignment and math operators combined)
+=
-=
*=
/=
%=

Example:

```
// assignment and math operators combined
            a += 5;   // is same as a = a + 5;
            Console.WriteLine("a is " + a);
            a -= 10;  // a = a - 10;
            Console.WriteLine("a is " + a);
            a *= 5;  // a = a * 5;
            Console.WriteLine("a is " + a);
```

How to get user's input?
Here is one way. We use Console.ReadLine(); to read user's text to a text variable. Then. If needed, we convert given data to proper data type.

```
Example
String text;
Console.WriteLine("Type your text");
text = Console.ReadLine();

// if you want to get numerical values, e.g. whole numbers:
String temp;
Console.WriteLine("Type your value");
temp = Console.ReadLine();
int realValue = Convert.ToInt16(temp);
```

Sometimes we have to make decisions in our apps: it is also called branching.
Now we discuss different choices for decision making.
Here below we have a list of relational operations that we often need to create conditional expressions that are the part of decision making…

Decision making (branching)

Program flow is depending on the condition. Relational operators (to create conditions) are explained here...

Operator	Explanation	Example

<	Smaller than	int a = 5;
		int b = 6;
		(a < b)
		true

<=	Smaller than or equal to	(a <= b) true
>	Bigger than	(a > b) false
>=	Bigger than or equal to	(a >= b) false
==	Equal to	(a == b) false
!=	Not equal to	(a != b) true

If statement

Syntax

```
if ( this is true )
    this code is executed
```

If several statements are executed after if, we have use program block!
Program block or a compound statement consists of zero or more statements enclosed in curly braces ({ }).

```
if ( this is true )
{
    this code is executed
    and this code is executed
    and …
}
```

```
We can have else part:
if ( this is true )
    this code is executed
else
   this code is executed


AND with several statements we can do like this
if ( this is true )
{
    this code is executed
    and this code is executed
    and …
}
else
{
    this code is executed
    and this code is executed
    and …
}
```

If statement: examples

Task 7

```
int a = 10;
int b = 20;
// is a smaller than b?
if (a < b)
    Console.WriteLine("a is smaller");

// is a bigger than b?
if (a > b)
    Console.WriteLine("a is bigger than b");

// is a equal to b?
if (a == b)
    Console.WriteLine("a is same as b");

// is a not equal to b?
if (a != b)
    Console.WriteLine("a is not same as b");

int x = 5;
int y = 6;
// check which of values is bigger, x or y
if ( x > y)
    Console.WriteLine("x is bigger");
else
    Console.WriteLine("y is bigger");

// tell the name of the value that is stored to variable z
int z = 5;
if (z == 5)
    Console.WriteLine("five");
```

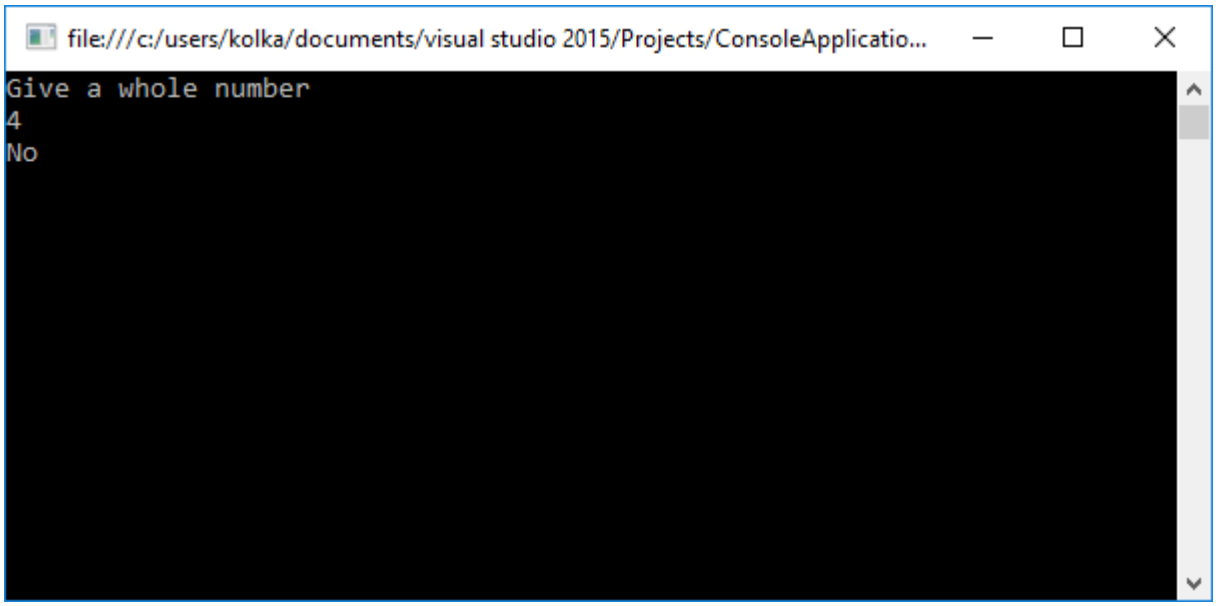
```
else  if (z == 4)
    Console.WriteLine("four");
else if (z == 3)
    Console.WriteLine("three");
else if (z == 2)
    Console.WriteLine("two");
else if (z == 1)
    Console.WriteLine("one");
else
    Console.WriteLine("do not know");

// is the value of p between 0 and 10?
int p = 5;
if (p >= 0)
    if (p <= 10)
        Console.WriteLine("Yes");
    else
        Console.WriteLine("No");
else
    Console.WriteLine("No");
```

Task 8

```
// User gives a value and our program tells if the value is >
 100 or not.
String answerAsText;
int asValue;
// ask user to give a value
Console.WriteLine("Give a whole number");
// take the value
answerAsText = Console.ReadLine();
// convert it to int
```

```
            asValue = Convert.ToUInt16(answerAsText);

            // bigger that 100?
            if (asValue > 100)
                Console.WriteLine("Yes");
            else
                Console.WriteLine("No");
```

Test run



Task 9

User enters a weekday number and the program tells the name of the day in Germany.

Solution

```
    String temp;
            int dayNr;
            Console.WriteLine("Give weekday number [1-7]");
            temp = Console.ReadLine();
            dayNr = Convert.ToInt16(temp);
            if (dayNr == 1)
                Console.WriteLine("Montag");
            else if (dayNr == 2)
                Console.WriteLine("Dienstag");
```

```
else if (dayNr == 3)
    Console.WriteLine("Mittwoch");
else if (dayNr == 4)
    Console.WriteLine("Donnerstag");
else if (dayNr == 5)
    Console.WriteLine("Freitag");
else if (dayNr == 6)
    Console.WriteLine("Samstag");
else if (dayNr == 7)
    Console.WriteLine("Sonntag");
else
    Console.WriteLine("Not a suitable weekday number!!");
```

Logical operators

Operator	Explanation	Example
&&	AND	int a = 5; (a >= 0 && a <=10) true
||	OR	(a < 0 || a > 10) false

Logical operators: Examples

```
//logical operators && and  || or
        int x = 50;
        // is x between 10 and 50 or not?
        if (x >= 10 && x <= 50)
            Console.WriteLine("Is between 10 and 50");
        else
            Console.WriteLine("Is NOT between 10 and 50");

        // doing the same with OR
        if (x < 10  || x > 50)
```

```
    Console.WriteLine("Is NOT between 10 and 50");
else
    Console.WriteLine("Is  between 10 and 50");
```

Task 10

Month number is given: program tells how many days that month has.

Solution

```
            30 days: 4, 6, 9, 11
            28/29: 2
            31: other months
            */
            int month = 10;
            if (month == 4 || month == 6 || month == 9 || month == 11)
                Console.WriteLine("30 days");
            else if
                (month == 1 || month == 3 || month == 5 || month == 7 ||
                 month == 8 ||
                 month == 10 || month == 12)
                Console.WriteLine("30 days");
            else if (month == 2)
                Console.WriteLine("28 days");
```

Task 11

Create a program: what is the biggest of 3 given values?

Solution

```
            int a, b, c;
            a = 50009;  b = 5557; c = 1000;
            // way 1
            if (a > b && a > c)
                Console.WriteLine("Biggest is a " + a);
            else if (b > a && b > c)
                Console.WriteLine("Biggest is b " + b);
            else
                Console.WriteLine("Biggest is c " + c);
            // way 2
            if (a > b)
```

```
            if (a > c)
                Console.WriteLine("Biggest is a " + a);
            else
                Console.WriteLine("Biggest is c " + c);
        else
            if (b > c)
               Console.WriteLine("Biggest is b " + b);
            else
               Console.WriteLine("Biggest is c " + c);
```

Task 12
User gives the lengths of the triangle's sides. Program tells what is the triangle like and calculates the area of the triangle.

We may have these triangletypes

Equilateral triangle

Isosceles triangle

Right angled triangle

Normal triangle

Solution

```
            int a, b, c;
            a = 4; b = 2; c = 3;
            if (a == b && a == c)
                Console.WriteLine("Equilateral!");
            else if (a ==b || a == c || b == c)
                Console.WriteLine("Isosceles!");
            else if (a*a + b*b == c*c || a*a + c*c == b*b || b*b + c*c ==
a*a)
                Console.WriteLine("Right angled!");
            else
                Console.WriteLine("Normal triangle");

            double s = (a + b + c) / 2.0;
```

```
        double area = Math.Sqrt(s * (s - a) * (s - b) * (s - c));
        Console.WriteLine("Area is " + area);

        Console.Read();
    }
```

Switch case

Is used when we have several choices
It is used instead of if-statements
It is said to be faster than if-statements
It is more readable

Switch case: examples

```
// example: variable stores a whole number between 0 and 5,
// and your program tells the number in German

int x = 4;
// with if first
if (x == 0)
    Console.WriteLine("null");
else if (x == 1)
    Console.WriteLine("eins");
else if (x == 2)
    Console.WriteLine("zwei");
else if (x == 3)
    Console.WriteLine("drei");
else if (x == 4)
    Console.WriteLine("vier");
else if (x == 5)
    Console.WriteLine("funf");
else
    Console.WriteLine("???");

// with switch case
switch (x)
{
    case 0: Console.WriteLine("null"); break;
    case 1: Console.WriteLine("eins"); break;
```

```
    case 2: Console.WriteLine("zwei"); break;
    case 3: Console.WriteLine("drei"); break;
    case 4: Console.WriteLine("vier"); break;
    case 5: Console.WriteLine("funf"); break;
    default: Console.WriteLine("???"); break;

}

// example: variable z contains a nr between 1 - 10 and
// program shows the value as a roman numeral
int z = 5;
switch (z)
{
    case 1:
        Console.WriteLine("I"); break;
    case 2:
        Console.WriteLine("II"); break;
    case 3:
        Console.WriteLine("III"); break;
    case 4:
        Console.WriteLine("IV"); break;
    case 5:
        Console.WriteLine("V"); break;
    case 6:
        Console.WriteLine("VI"); break;
    case 7:
        Console.WriteLine("VII"); break;
    case 8:
        Console.WriteLine("VII"); break;
    case 9:
        Console.WriteLine("IX"); break;
    case 10:
```

```
        Console.WriteLine("X"); break;
    default: Console.WriteLine("???"); break;

}


// example: character is printed as greek letter
// a, b, e, k, t, o
char letter = 'a';
switch (letter)
{
    case 'a': Console.WriteLine("alpha"); break;
    case 'b': Console.WriteLine("beta"); break;
    case 'e': Console.WriteLine("epsilon"); break;
    case 'k': Console.WriteLine("kappa"); break;
    case 't': Console.WriteLine("tau"); break;
    case 'o': Console.WriteLine("omikron"); break;
    default: Console.WriteLine("not known"); break;
}
```

Loops

We use loops for repeating some part of the code until some solution is found

A bit about program flow:
In programs execution flow can
a) go on straight forward (step by step)
b) contain decision making (branching)
c) contain loops

Examples of usages of loops:
when searching for a value from an array
when generating and printing hundreds of random numbers
in iterations

There are mainly two kinds of loops: for loop (when code is to be repeated fixed nnumber of time) and while loop (called often conditional loop).

for-loop

syntax
for (initialization of the counter variable; condition when to stop; functional part increment the counter)

{

code that is executed several times

}

Examples

```
// for loop
            //Example: print out values 1 to 10.
            for (int k = 1;  k <= 10; k = k + 1)
            {
                Console.WriteLine("k is  " + k);
            }

            // Example: print out 4, 8, 12, … 24
            for (int k = 4; k <= 24; k = k + 4)
```

```
{
    Console.WriteLine("k is  " + k);
}

// Example: print out values 0, 0.3, 0.6, ... 3
for (double i = 0; i <= 3; i = i + 0.3)
{
    Console.WriteLine("i is  " + i);
}

// example: print out value 10,8,6,...0
for (int k = 10; k >= 0; k = k - 2)
{
    Console.WriteLine("k is  " + k);
}

// example: calc. the sum of values 1.5, 3, 4.5, ... 20
double sum1 = 0;
int amount = 0;
for (double i = 1.5; i <= 20; i = i + 1.5)
{
    sum1 = sum1 + i;
    amount = amount + 1;
}

// example: calc. even the average of prev. example
double average = sum1 / amount;

Console.WriteLine("sum is  " + sum1);
Console.WriteLine("average is  " + average);
```

```
// for loop
            //Example: print out values 1 to 10.
            for (int k = 1;  k <= 10; k = k + 1)
            {
                Console.WriteLine("k is  " + k);
            }

            // Example: print out 4, 8, 12, … 24
            for (int k = 4; k <= 24; k = k + 4)
            {
                Console.WriteLine("k is  " + k);
            }

            // Example: print out values 0, 0.3, 0.6, ... 3
            for (double i = 0; i <= 3; i = i + 0.3)
            {
                Console.WriteLine("i is  " + i);
            }

            // example: print out value 10,8,6,...0
            for (int k = 10; k >= 0; k = k - 2)
            {
                Console.WriteLine("k is  " + k);
            }

            // example: calc. the sum of values 1.5, 3, 4.5, ... 20
            double sum1 = 0;
            int amount = 0;
            for (double i = 1.5; i <= 20; i = i + 1.5)
            {
                sum1 = sum1 + i;
```

```
            amount = amount + 1;
        }

        // example: calc. even the average of prev. example
        double average = sum1 / amount;

        Console.WriteLine("sum is  " + sum1);
        Console.WriteLine("average is  " + average);
```

About random numbers shortly

How to get random numbers?

Random object is needed:

Random generator = new Random();

Example: getting values between 1 and 10:

```
 int value = generator.Next(1,11);
```

```
        // example: let's generate 100 random numbers that are
between 0 to 20
        for (int k = 0; k < 100; k = k + 1)
        {
            int x2 = r.Next(0, 21);
            Console.Write(" " + x2);
        }

        // example: program throws dice 1000 times and counts amounts
    of 1's, 2's and so on...
      int n1, n2, n3, n4, n5, n6;
        n1 = 0; n2 = 0; n3 = 0;
      n4 = 0; n5 = 0; n6 = 0;
        for (int k = 0; k < 1000; k = k + 1)
        {
            int diceNumber = r.Next(1, 7);
```

```
            switch (diceNumber)
            {
                case 1: n1 = n1 + 1; break;
                case 2: n2 = n2 + 1; break;
                case 3: n3 = n3 + 1; break;
                case 4: n4 = n4 + 1; break;
                case 5: n5 = n5 + 1; break;
                case 6: n6 = n6 + 1; break;
            }
        }
        Console.WriteLine();
        Console.WriteLine("1: " + n1);
        Console.WriteLine("2: " + n2);
        Console.WriteLine("3: " + n3);
        Console.WriteLine("4: " + n4);
        Console.WriteLine("5: " + n5);
        Console.WriteLine("6: " + n6);
```

Conditional loops

while loop

Syntax:

```
while (condition is true)
{
code
code
};
```

do – while loop

Syntax:

```
do
{
code
code
}
while (condition is true);
```

Examples

```
    // example: program prints out values 2,4, ..,20
            // use a) for b) while c) do-while

            // for
            for (int k = 2; k <= 20; k = k + 2)
            {
                Console.WriteLine("" + k);
            }

            // while
            int i = 2;
            while (i <= 20)
            {
                Console.WriteLine("" + i);
                i = i + 2;
            }

            // do while
            int j = 2;
            do
            {
                Console.WriteLine("" + j);
                j = j + 2;
            }
            while (j <= 20);
```

Break and continue statements

Used with loops
Note: break was used even with switch-case

With break
you can terminate the loop when some condition is true

E.g.
When searching for a value from an array by using a loop:
when the value is found, there no use to go on searching,
just terminate the loop with break statement!

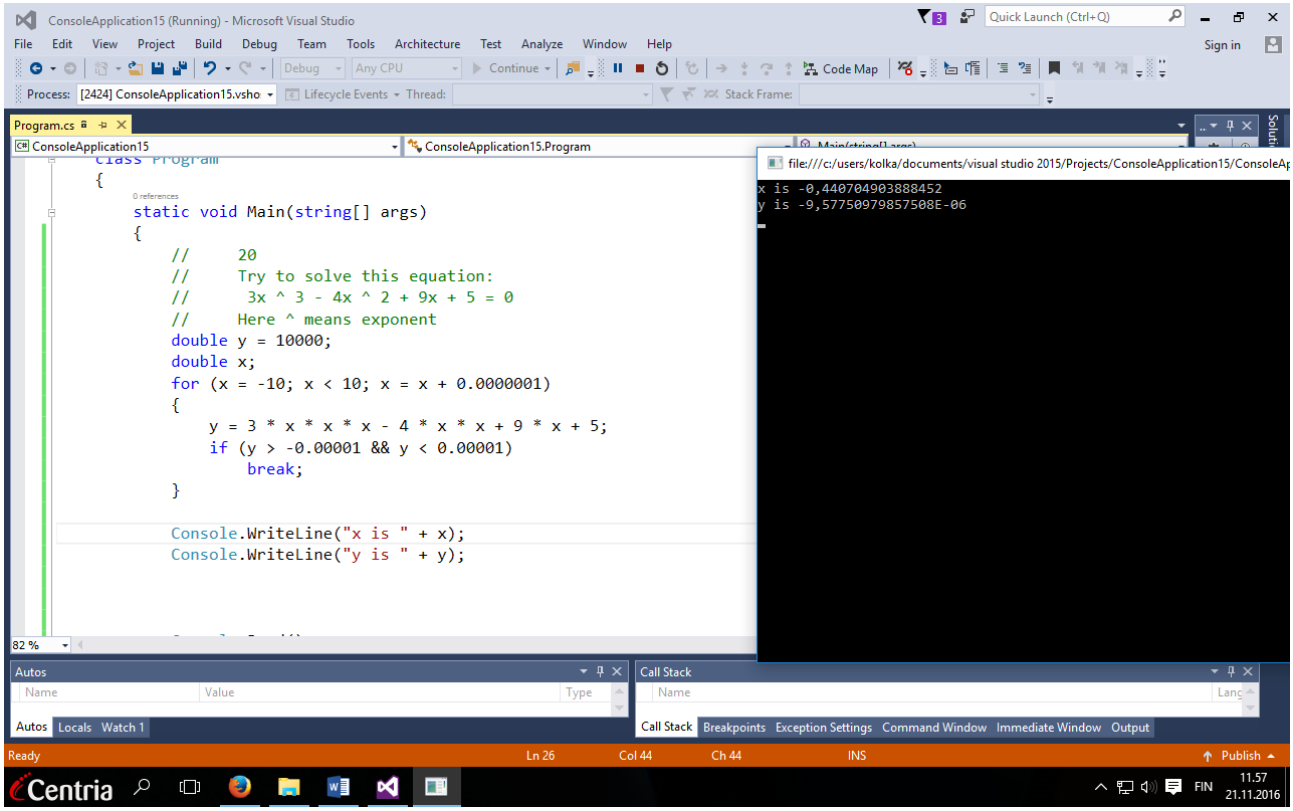
With continue
you can start a new round without executing the code that exists below continue statement

Example of break:

```
// break
            // print 10 members of series 3, 9, 27, 81, ..
            int n = 0;
            for (long k = 3; k < 200000000000; k = k * 3)
            {
                Console.Write(" " + k);
                n = n + 1;
                if (n == 10)
                    break;
            }


            Console.Read();

            // Example: Try to solve this equation:
            //      3x ^ 3 - 4x ^ 2 + 9x + 5 = 0
            //     Here ^ means exponent
            double y = 10000;
            double x;
            for (x = -10; x < 10; x = x + 0.0000001)
            {
```

```
    y = 3 * x * x * x - 4 * x * x + 9 * x + 5;
    if (y > -0.00001 && y < 0.00001)
        break;
}

Console.WriteLine("x is " + x);
Console.WriteLine("y is " + y);
```

Here is also a screen copy about the test run.



Arrays

Normal variable can store only 1 value.
Array can store several values of same data type.

One dimensional array

Example: an array that can store 5 integers

20	9	-55	3	123

```
int[] values = new int[5];

values[0] = 20;
values[1] = 9;
values[2] = -55;
values[3] = 3;
values[4] = 123;

Initialization:
E.g. our array will contain values 1.5, 2.5, 5.0, 10.9
double[] measures = {1.5, 2.5, 5.0, 10.9};
```

Example: Array stores USA president names:

```
String[] names = new String[5];
names[0] = “Lincoln”;
names[1] = “Reagan”;
names[2] = “Kennedy”;
names[3] = “Eisenhower;
names[4] = “Trump”;
```

2 dimensional array

Has rows and columns.

Germany	Sweden	Finland
Berlin	Stockholm	Helsinki

```
String[,] countries = new String[2, 3];
countries[0,0] = “Germany”;
countries[0,1]  = “Sweden”;
countries[0,2]  = “Finland”;
```

```
countries[1,0] = “Berlin”;
countries[1,1] = “Stockholm”;
countries[1,2] = “Helsinki”;
```

How to initialize a 2 dimensional array?

```
String[,] earnings = { {“May”, “1000”},
                       {“June”,”2000”},
                       {“July”, “1990”},
     {“August”, “3000} };
```

ARRAY problems & solutions

Here we take a look at some general array problems:

Filling an array with random numbers

Printing out array values.

Calculate the sum and the average.

And also these common problems:

Searching for the minimum/maximum value

Checking if a specific value is in the table

Sorting an array

Minimun and maximun:
Principle:
Assign the first value of the table to some variable (e.g named min or max):
Then we check if there are smaller or bigger values in the remaining part of the array
If bigger or smaller value is found, it is assigned to min or max variable

// array problems:

```
// filling an array first
            int[] values = new int[10];
            Random rand = new Random();
            for (int k = 0; k < 10; k = k + 1)
            {
                values[k] = rand.Next();
```

```
            }

// printing an array
            for (int k = 0; k < 10; k = k + 1)
            {
                Console.WriteLine(" " + values[k]);
          }

// searching for min value
            // minimum
            int min = values[0];
            for (int k = 1; k < 10; k = k + 1)
          {
                if (values[k] < min)
                    min = values[k];
          }

            Console.WriteLine("Minimun value is " + min);

  // searching for max value
            int max = values[0];
            for (int k = 1; k < 10; k = k + 1)
            {
                if (values[k] > max)
                    max = values[k];
            }

            Console.WriteLine("Maximum value is " + max);
```

Searching for a specific value:
Principle:
Inside a loop we start comparing the value of array to the value we are searching for
If values are same

add the position to some variable
break the loop (no use to go on…)

After loop we can test the variable: if it has some positive value,
we can print that value was found
else
we print that it was not found

```
// searching for a specific value
            int x = 1234;

            values[5] = x;

            int place = -1;
            for (int k = 0; k < 10; k = k + 1)
            {
                if (values[k] == x)
                {
                    place = k;
                    break;
                }
            }

            if (place == -1)
                Console.WriteLine("not found :(");
            else
                Console.WriteLine("value was found, the place is " +
place);
```

Supporting classes

Many programming languages have labraries, packages or classes that contain many useful functions that help programmers!!

Here is one set of classes that C# codes can use: Math, Convert, Console, Random, String

Let's take a look at those treasures!!

Math, Convert, Console, Random, String

Examples tell you best how to use them: we have already used Convert, Console, and Random classes. So we concentrate on Math and String.

```
// Math class
            // Value of pi and e
            Console.WriteLine("Pi is " + Math.PI);
            Console.WriteLine("Value of e is " + Math.E);
            // square root
            double x = 5;
            double result = Math.Sqrt(x);
            Console.WriteLine("Result is " + result);
            // exponentiation
            result = Math.Pow(x, 3);
            Console.WriteLine("Result is " + result);

            // floor and ceiling functions
            result = Math.Floor(6.9);
            Console.WriteLine("Result is " + result);

            result = Math.Ceiling(7.2);
```

```
            Console.WriteLine("Result is " + result);

            // radians and degrees
            double degrees = 50;
            // radians = degrees × PI / 180°
            double radians = degrees * Math.PI / 180;
            Console.WriteLine("radians: " + radians);

            // degrees = radians * 180/PI
            degrees = radians * 180 / Math.PI;
            Console.WriteLine("degrees: " + degrees);

            // angle is 50 degrees, what is sine()
            // 50 degrees is 0.87 radians
            result = Math.Sin(0.87);
            Console.WriteLine("Result is " + result);

// String
            String name = "China";
            int size = name.Length;
            Console.WriteLine("size is " + size);

            String part1 = name.Substring(0, 3);
            Console.WriteLine("string is " + part1);

            // position of some letter
            int place = name.IndexOf('i');
            Console.WriteLine("" + place);

            // what letter is in that place?
            char letter = name.ElementAt(4);
            Console.WriteLine("" + letter);
```

```
// trimming
String city = "    Helsinki  ";
size = city.Length;
Console.WriteLine("size is " + size);

city = city.Trim();
size = city.Length;
Console.WriteLine("size is " + size);
```

Now we move to a new topic: functions!!

It is very important topic: let's go on!

Functions

Function are also often called subprograms, routines, procedures or methods.

How do functions help programmer and what is a function?

Function:
Does one well defined task
Can be called several times from other part of the program
Can be reused in other programs

So, program is better organized (better structure)
No need for repeating same code
When some code is to be used more than once, it is good to create a function

Function definition (C#)

Access specifier return type nameOfTheFunction(possible parameters)
{
  body of the function (the code)
}

If return type is **void**, function does not return anything!

In C# we use keyword static for now…

Note: We have already used one function, Main-function!!

Examples of functions

```
//Our function prints out “Good Morning”
static void sayGreeting()
{
    Console.WriteLine("Good Morning");
}

// Our function prints out our greeting
static void sayThisGreeting(String greeting)
{
    Console.WriteLine(greeting);
}

//  Our function prints out our greeting n times
static void sayThisSeveralTimes(String mess, int n)
{
    for (int k = 0; k < n; k++)
    {
        Console.WriteLine(mess);
    }
}

// Our function returns the sum of 3 integers
static int sumOfThree(int a, int b, int c)
{
    int n = a + b + c;
    return n;  // return the result to the caller
}

// Our function returns the perimeter of a circle when
// radius (a whole number) is passed to the function
static double perimeter(int r)
```

```
  {
      return (2 * 3.14 * r);
  }
  static void Main(string[] args)
  {
      // call
      sayGreeting();
      sayThisGreeting("Good Evening");
      sayThisSeveralTimes("Hello", 10);
      // way 1
      Console.WriteLine("The sum is {0} ", sumOfThree(8, 9, 10));
      // way 2
      int x = sumOfThree(33, 44, 55);
      Console.WriteLine("The sum is {0} ", x);
      // way 3
      int p1 = 20; int p2 = 100; int p3 = 200;
      x = sumOfThree(p1, p2, p3);
      Console.WriteLine("The sum is {0} ", x);

      Console.WriteLine("The perimeter is {0} ", perimeter(3));
      Console.Read();
}
```

Passing an array to the function

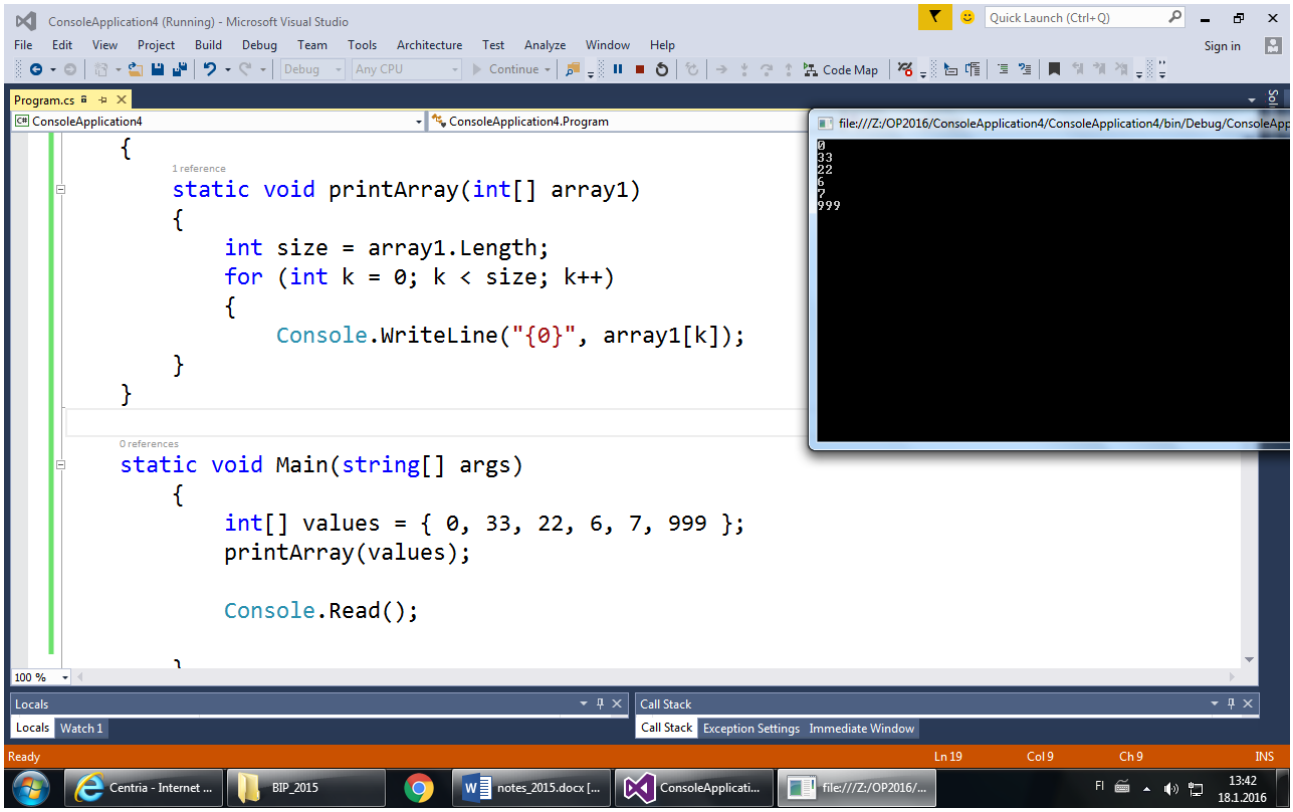
Note: array is passed as a reference – it meant that the function can make permanent changes to the array!

Example: The function prints the values of an array

```
static void printArray(int[] array1)
{
    int size = array1.Length;
    for (int k = 0; k < size; k++)
    {
     Console.WriteLine(“{0}”, array1[k]);
    }
}
```

```
How to call this function?
int[] values = {0,  33, 22, 6,7,999};
printArray(values);
```

Here is also a screen copy about the rest run:



Examples: arrays and functions

```
        // printing array
         static void printArray(int[] arr)
         {
             int size = arr.Length;
             for (int k = 0; k < size; k++)
                 Console.Write("{0} ", arr[k]);
         }

         // dublicate values
         static void multiplyByTwo(int[] arr)
         {
             int size = arr.Length;
             for (int k = 0; k < size; k++)
                 arr[k] = arr[k] * 2;

         }

        // return average
         static double average(int[] arr)
         {
             int sum = 0;
             int size = arr.Length;
             for (int k = 0; k < size; k++)
                 sum = sum + arr[k];

             double aver = (double) sum / size;

             return aver;
         }
```

```
 static void Main(string[] args)
{
    int[] values = { 11, 22, 3, 4 };
    printArray(values);
    multiplyByTwo(values);
  // NOTE: arrays are automatically passed by reference
    printArray(values);
    double x = average(values);
    Console.WriteLine("Average is {0}", x);
    Console.Read();
}
```

More about function parameters

Passing by value AND passing by reference: what does it mean?

When passing by value: method can not change the original variable: only the copy of the value is passed to the method.

When passing by reference, the address of the variable is passed to the method and method can manipulate that memory place.

See example here:

```
        static void add5(int a)
        {
            a += 5;
        }

        static void add5really(ref int a)
        {
            a += 5;
        }
        static void Main(string[] args)
          {
            int x = 10;
            Console.WriteLine("x is now " + x);

            add5(x);
            Console.WriteLine("x is now " + x);  // prints 10

            add5really(ref x);
            Console.WriteLine("x is now " + x);  // prints 15!!
}
```

Note:

This was part 1: basics of programming and creating console applications.

But: we often want to create a better user interface to our apps! That's why we have to know the concept graphical user interface and how to create it. What kinds of components are there? How do I know if user has for example clicked a button by mouse? How does my code can be assigned to that mouseclick?

There are realy several questions!!

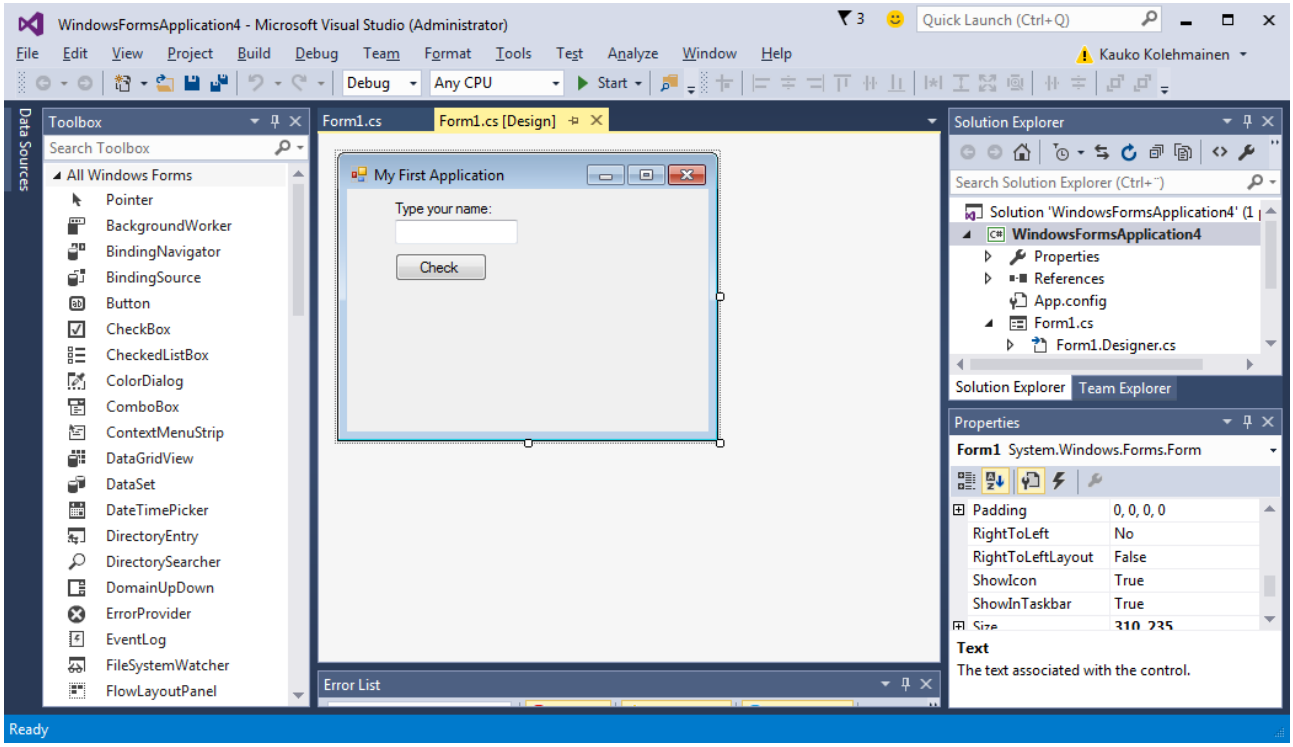
But now we move to that part, part 2 that teaches you to create GUIs!!!

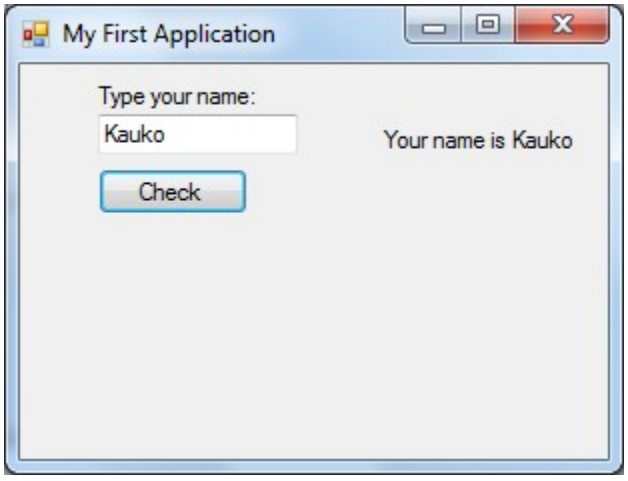
GUI

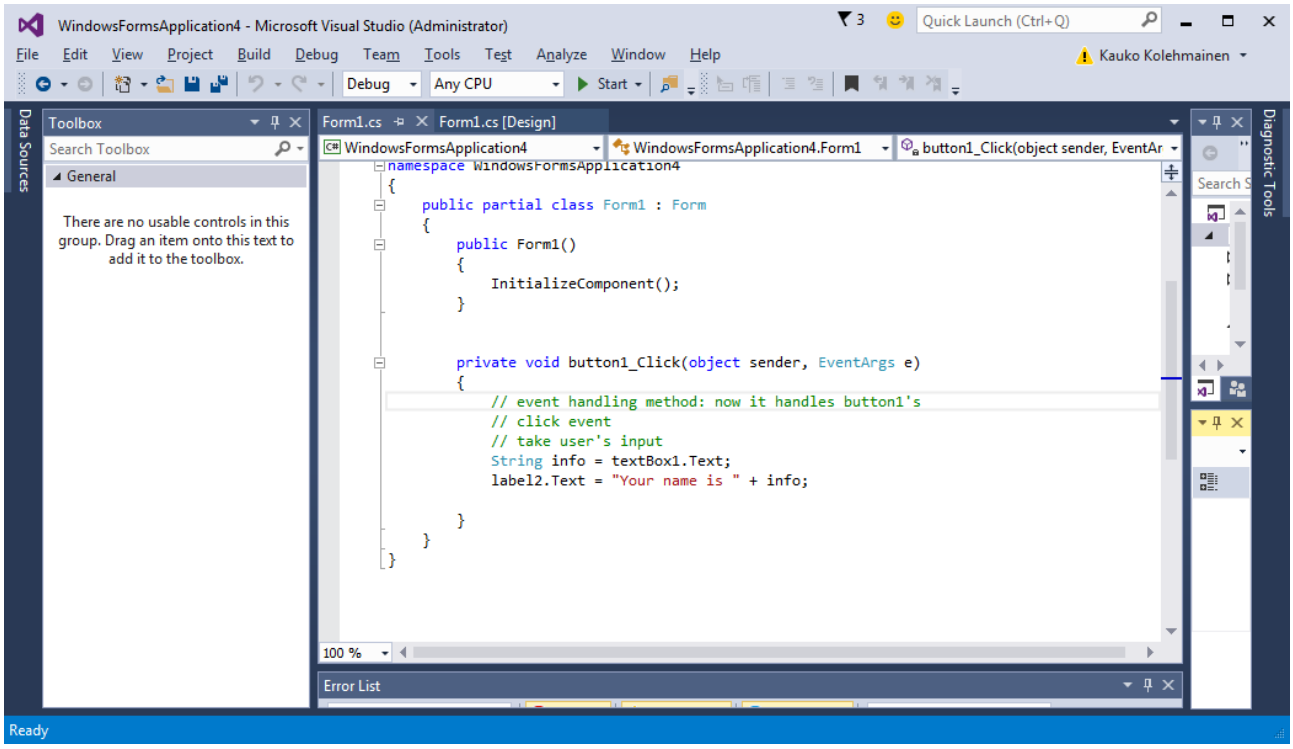
Now we create some smaller applications that help to understand GUI applications!

First you have to start a new project that is a windows form application.

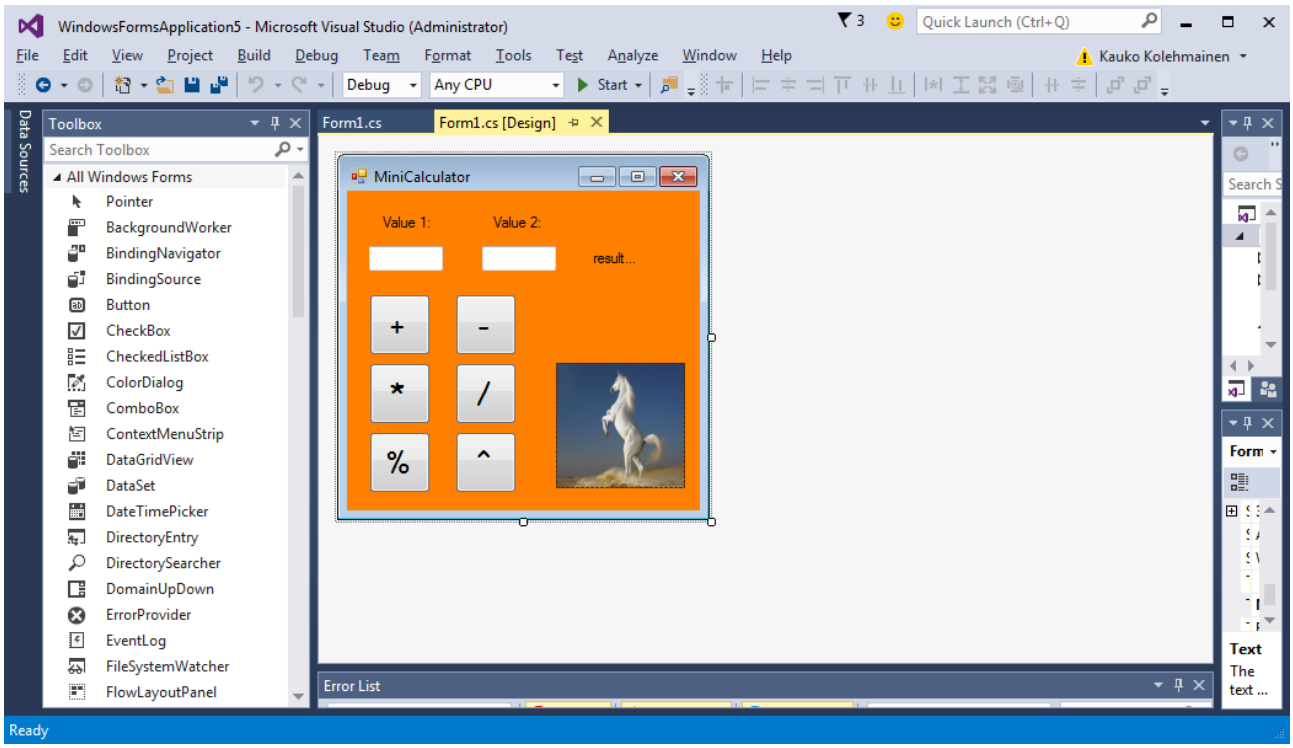
Video: starting a gui project and introduction of the Visual Studio interface.

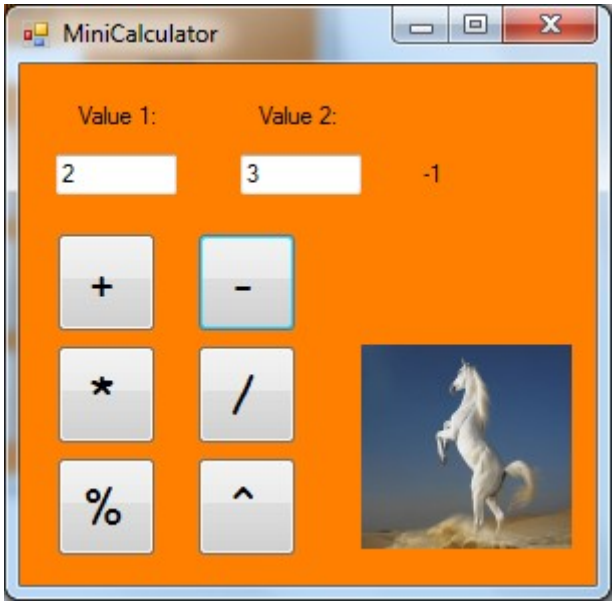






Mini Calculator





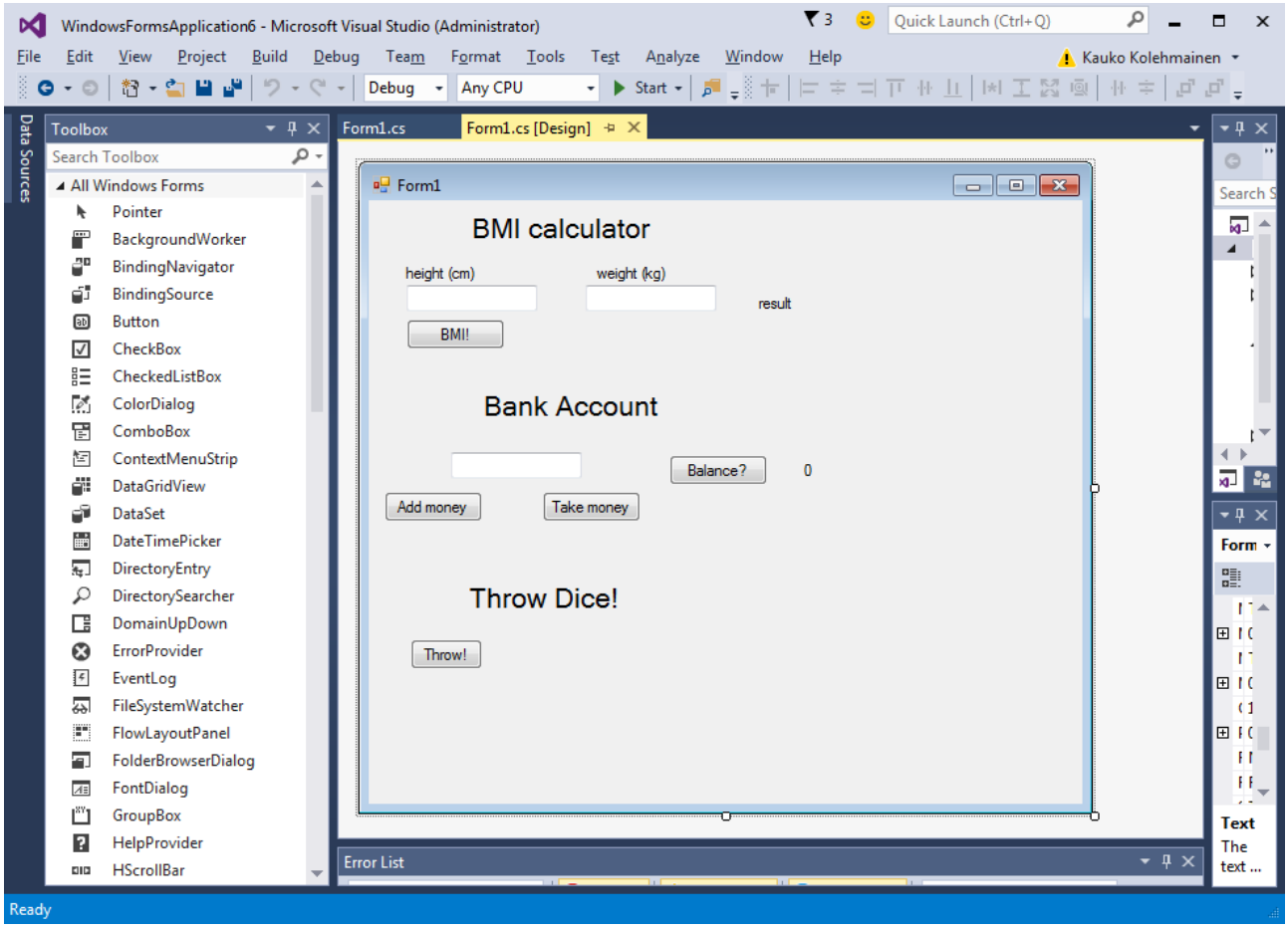

```
private void button1_Click(object sender, EventArgs e)
        {
            String temp1, temp2;
            temp1 = textBox1.Text;
            temp2 = textBox2.Text;

            int a, b, result;
            a = Convert.ToInt32(temp1);
            b = Convert.ToInt32(temp2);

            result = a + b;

            label3.Text = "" + result;
        }

        private void button2_Click(object sender, EventArgs e)
        {
            String temp1, temp2;
            temp1 = textBox1.Text;
            temp2 = textBox2.Text;
```

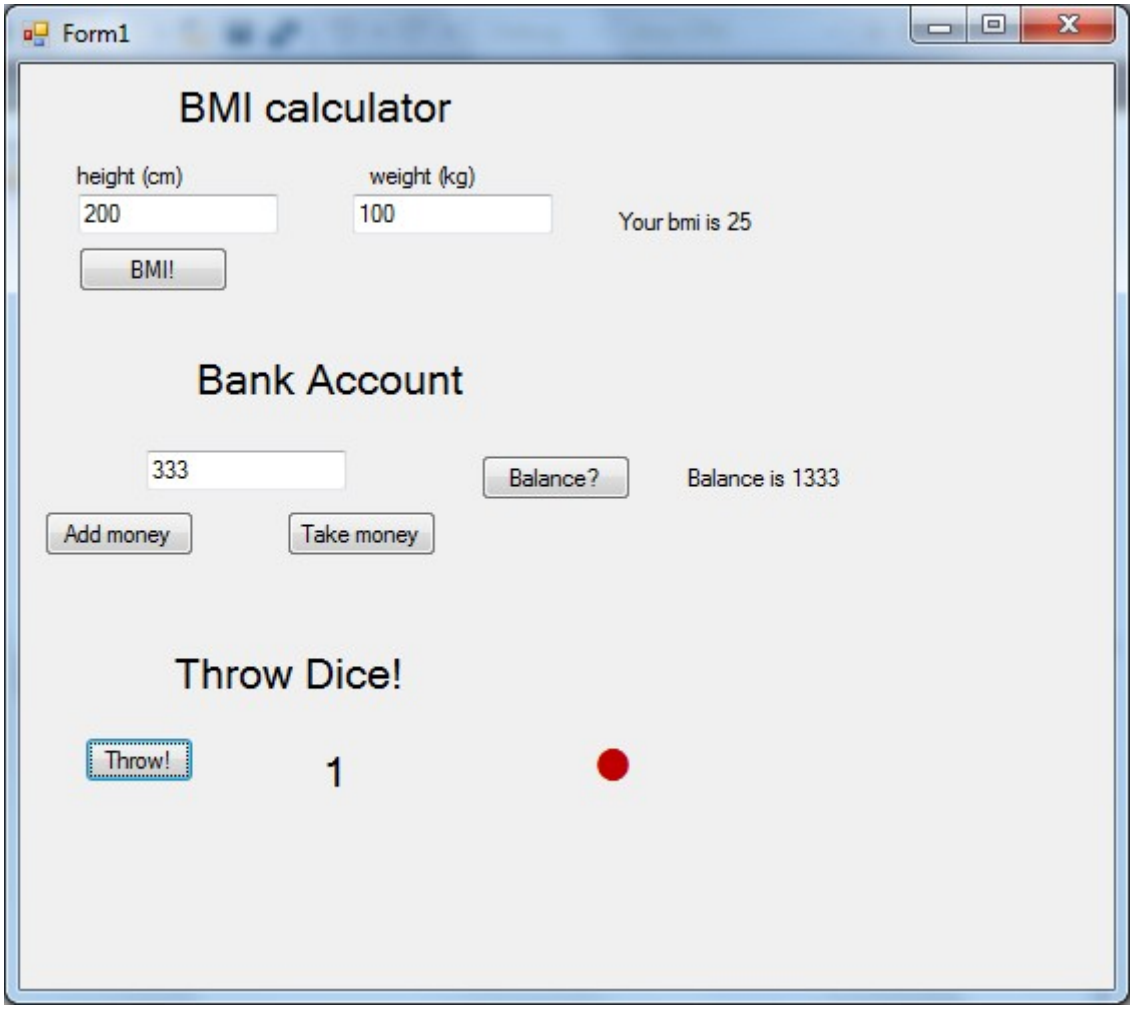
```
    int a, b, result;
    a = Convert.ToInt32(temp1);
    b = Convert.ToInt32(temp2);

    result = a - b;

    label3.Text = "" + result;
}

private void button3_Click(object sender, EventArgs e)
{
    String temp1, temp2;
    temp1 = textBox1.Text;
    temp2 = textBox2.Text;

    int a, b, result;
    a = Convert.ToInt32(temp1);
    b = Convert.ToInt32(temp2);

    result = a * b;

    label3.Text = "" + result;

}

private void button4_Click(object sender, EventArgs e)
{
    String temp1, temp2;
    temp1 = textBox1.Text;
    temp2 = textBox2.Text;
```

```
    int a, b;
    double result;
    a = Convert.ToInt32(temp1);
    b = Convert.ToInt32(temp2);

    result = (double) a / b;

    label3.Text = "" + result;


}

private void button5_Click(object sender, EventArgs e)
{
    String temp1, temp2;
    temp1 = textBox1.Text;
    temp2 = textBox2.Text;

    int a, b, result;
    a = Convert.ToInt32(temp1);
    b = Convert.ToInt32(temp2);

    result = a % b;

    label3.Text = "" + result;

}

private void button6_Click(object sender, EventArgs e)
{
    String temp1, temp2;
    temp1 = textBox1.Text;
```

```
    temp2 = textBox2.Text;

    int a, b;
    double result;
    a = Convert.ToInt32(temp1);
    b = Convert.ToInt32(temp2);

    result = Math.Pow(a, b);

  label3.Text = "" + result;

}
```

BMI Calculator



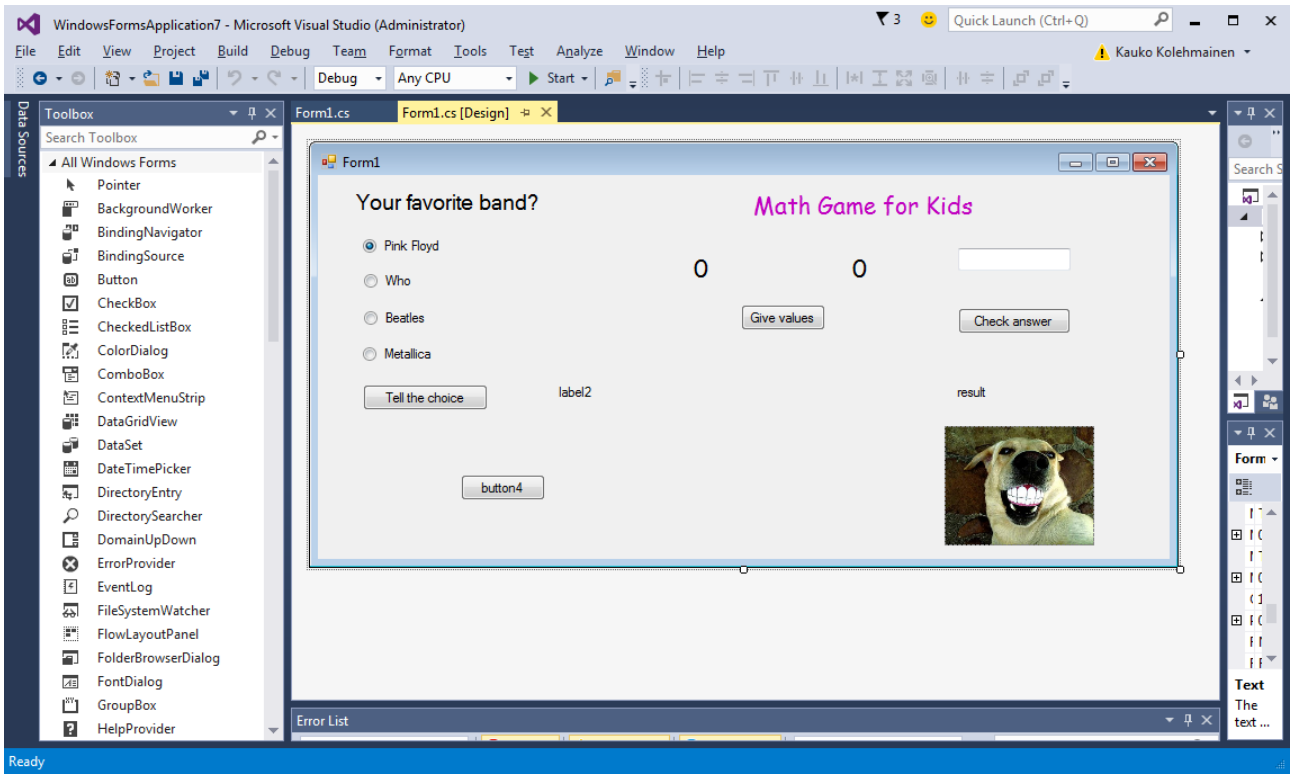


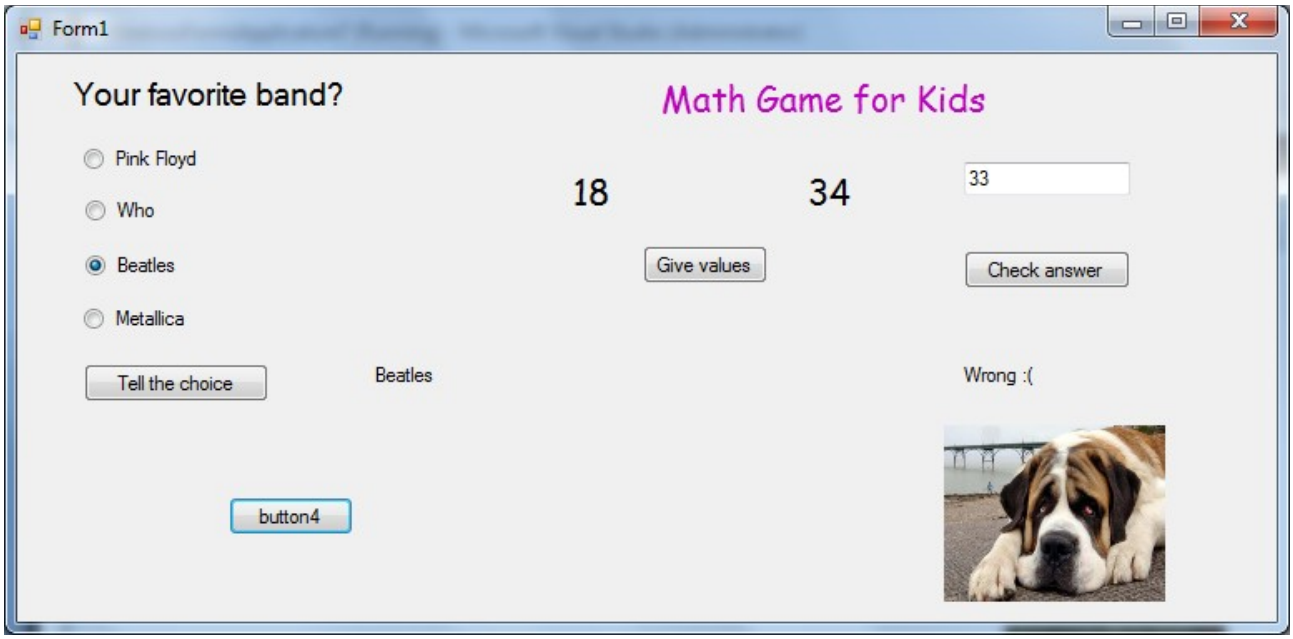

```
private void button1_Click(object sender, EventArgs e)
      {
          String a, b;
          a = textBox1.Text;
          b = textBox2.Text;

          double h = Convert.ToDouble(a);
          double w = Convert.ToDouble(b);

          h = h / 100;  // height to meters

          double bmi = w / (h * h);
          bmi = Math.Round(bmi, 2);
```

```
    label4.Text = "Your bmi is " + bmi;


}

double balance = 1000;  // global storage needed!
private void button2_Click(object sender, EventArgs e)
{
    String temp = textBox3.Text;
    double sum = Convert.ToDouble(temp);
    balance = balance + sum;
    label6.Text = "Balance is " + balance;
}

private void button3_Click(object sender, EventArgs e)
{
    String temp = textBox3.Text;
    double sum = Convert.ToDouble(temp);
    if (balance < sum)
        label6.Text = "You can not take so much!! :)";
    else
    {
        balance = balance - sum;
        label6.Text = "Balance is " + balance;
    }
}

private void button4_Click(object sender, EventArgs e)
{
    label6.Text = "Balance is " + balance;
}
```

```
private void button5_Click(object sender, EventArgs e)
{
    Random generator = new Random();
    int nr = generator.Next(1, 7);
    label8.Text = "" + nr;

    if (nr == 1)
        label9.Text = "l";
    if (nr == 2)
        label9.Text = "ll";
    if (nr == 3)
        label9.Text = "lll";
    if (nr == 4)
        label9.Text = "llll";
    if (nr == 5)
        label9.Text = "lllll";
    if (nr == 6)
     label9.Text = "llllll";


}
```

Math Game






```
private void button1_Click(object sender, EventArgs e)
        {
            String choice = "";
```

```
    if (radioButton1.Checked == true)
        choice = radioButton1.Text;
    if (radioButton2.Checked == true)
        choice = radioButton2.Text;
    if (radioButton3.Checked == true)
        choice = radioButton3.Text;
    if (radioButton4.Checked == true)
        choice = radioButton4.Text;

    label2.Text = choice;

}

int a, b, result;
int corrects = 0;
int wrongs = 0;

private void button4_Click(object sender, EventArgs e)
{
    System.Media.SoundPlayer audio1;
    audio1 = new System.Media.SoundPlayer();
    audio1.SoundLocation = "z:\\pics\\sound2.wav";
    audio1.Load();
    audio1.Play();

}

private void button3_Click(object sender, EventArgs e)
{
    result = Convert.ToInt16(textBox1.Text);
    if (result == (a + b))
    {
```

```
            corrects++;
            label6.Text = "Right :)";
            pictureBox1.Image = Image.FromFile("z:\\pics\\dog1.jpg");
            pictureBox1.Visible = true;
        }
        else
        {
            wrongs++;
            label6.Text = "Wrong :(";
            pictureBox1.Image = Image.FromFile("z:\\pics\\dog2.jpg");
            pictureBox1.Visible = true;
        }


    }


    private void button2_Click(object sender, EventArgs e)
    {
        Random rr = new Random();
        a = rr.Next(10, 51);
        b = rr.Next(10, 51);

        label4.Text = "" + a;
        label5.Text = "" + b;
        pictureBox1.Visible = false;

}
```

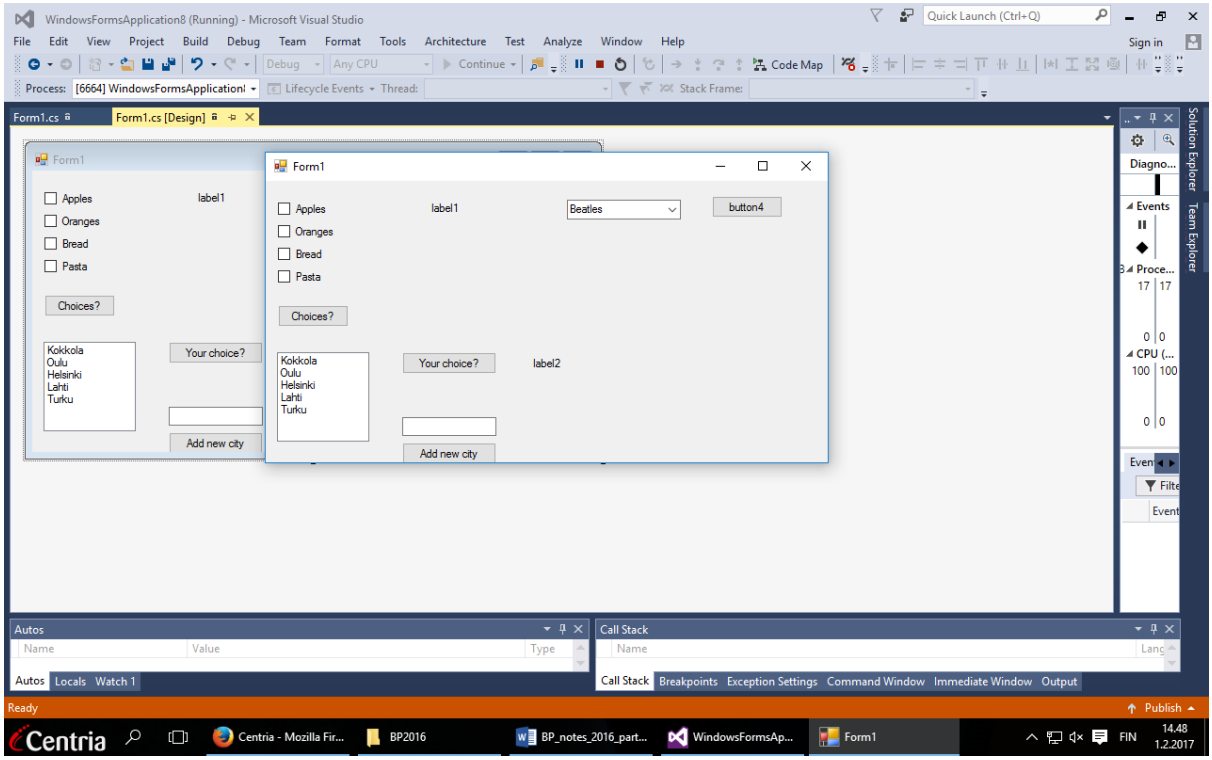
Checkboxes

1 or more of them can be chosen

Example:

```
String what = "";
private void button1_Click(object sender, EventArgs e)
{
    if (checkBox1.Checked == true)
        what += checkBox1.Text + " ";
    if (checkBox2.Checked == true)
        what += checkBox2.Text + " ";
    if (checkBox3.Checked == true)
        what += checkBox3.Text + " ";
    if (checkBox4.Checked == true)
        what += checkBox4.Text + " ";

    label1.Text = what;
}

private void button2_Click(object sender, EventArgs e)
{
    label2.Text = listBox1.Text;
}

private void button3_Click(object sender, EventArgs e)
{
    String item = textBox1.Text;

    if (listBox1.Items.Contains(item))
        label2.Text = "Is there already!";
```

```
        else
            listBox1.Items.Add(item);

    }

    private void Form1_Load(object sender, EventArgs e)
{
        comboBox1.SelectedIndex = 0;
    }
```



Examples and special things

Creating controls dynamically

Here we first create 2 labels dynamically:

Create a new project.

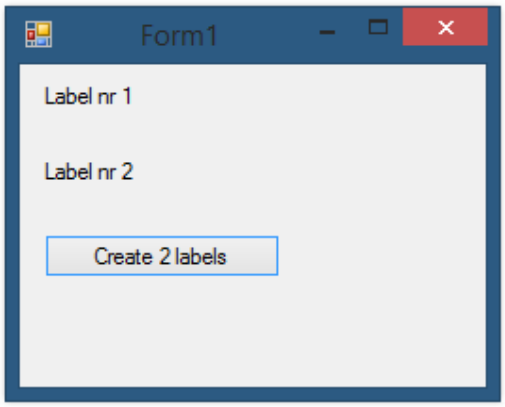
Add there a button with caption "Create labels".

When button is clicked 2 labels are created and added to the form.

So we have to:
define variables (object) of type Label
set their widths and heights
add their captions: "Label nr 1" and "Label nr 2"
set their parent (now the form is their parent)
set their location

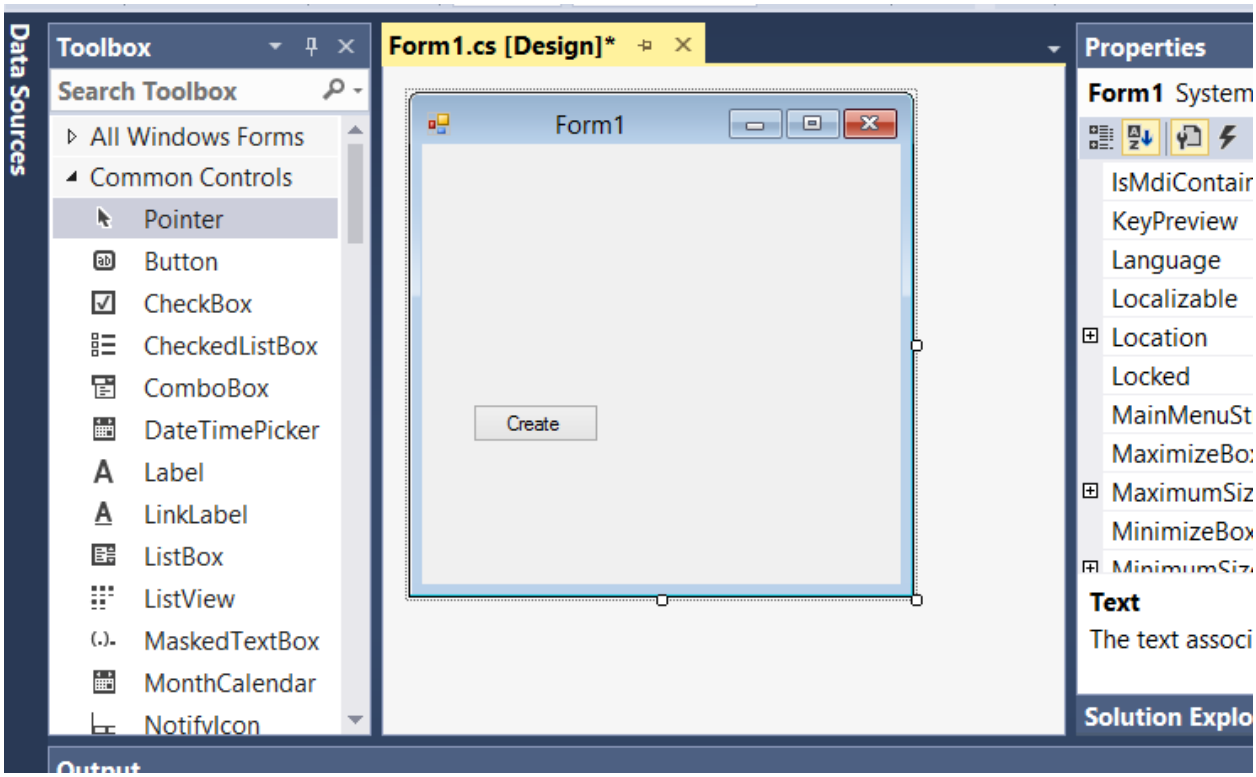
SO write this code:

```
    Label L1, L2;
    private void button1_Click(object sender, EventArgs e)
    {
        // create label objects
        L1 = new Label();
        L2 = new Label();

        // set sizes
        L1.Height = 20;
        L1.Width = 100;
        L2.Height = 20;
        L2.Width = 100;

        // add captions
        L1.Text = "Label nr 1";
        L2.Text = "Label nr 2";

        // set parents
        L1.Parent = this;   // this refers to the form
        L2.Parent = this;

        // add locations
        L1.Top = 10;
        L1.Left = 10;

        L2.Top = 50;
        L2.Left = 10;


    }
}
```

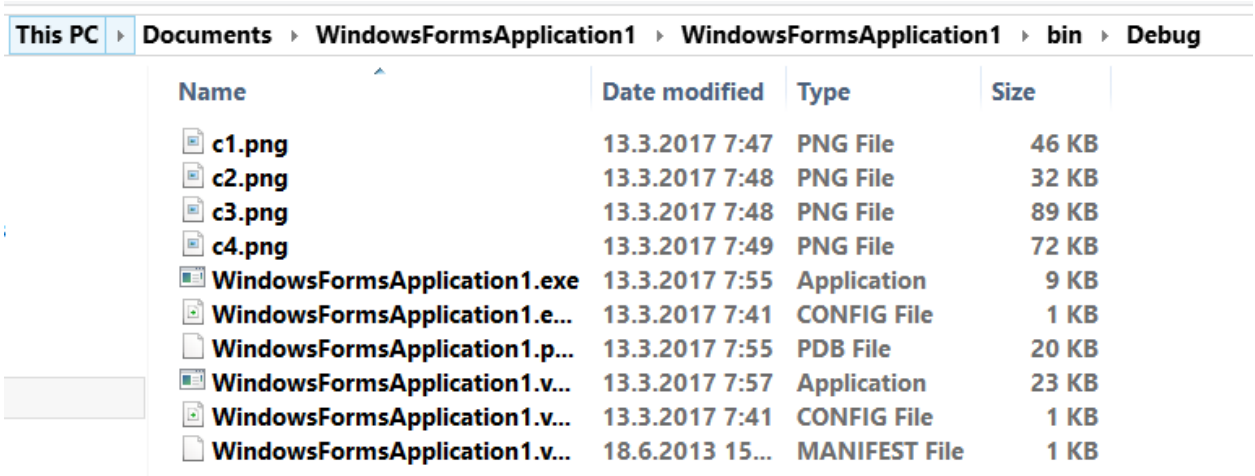


Lets create 4 pictureboxes and add them to random places to the form:

Start a new project.



Add some pictures, here pics are added to the Debug-subfolder of the project:

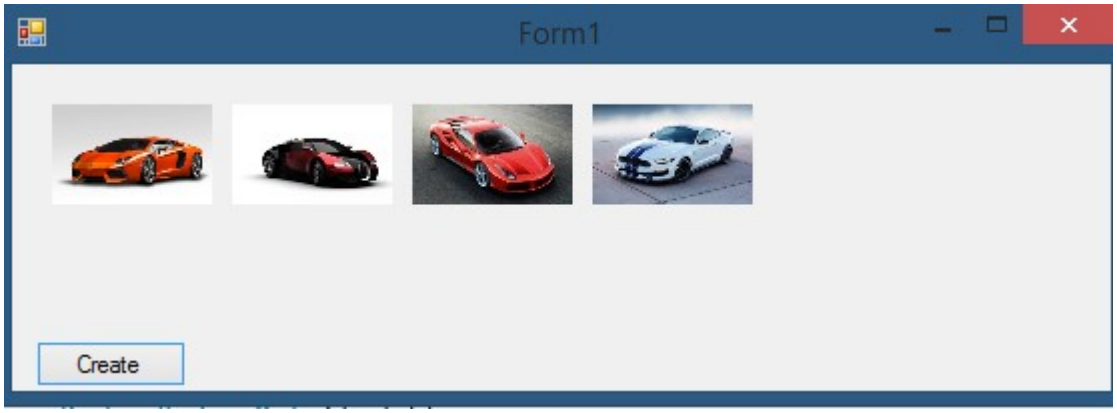


Write the code

```
private void button1_Click(object sender, EventArgs e)
        {
            // create controls
            p1 = new PictureBox();
            p2 = new PictureBox();
            p3 = new PictureBox();
            p4 = new PictureBox();

            // add sizes
            p1.Width = 80;
            p1.Height = 50;
            p2.Width = 80;
            p2.Height = 50;
            p3.Width = 80;
            p3.Height = 50;
            p4.Width = 80;
            p4.Height = 50;

            // Add to the form
            p1.Parent = this;
```

```
            p2.Parent = this;
            p3.Parent = this;
            p4.Parent = this;



            // Add images to pictureboxes
            p1.Image = Image.FromFile("c1.png");  // now pictures can be
found from Debug
            p2.Image = Image.FromFile("c2.png");
            p3.Image = Image.FromFile("c3.png");
            p4.Image = Image.FromFile("c4.png");

            // change sizemodes when needed, e.g:
            p1.SizeMode = PictureBoxSizeMode.StretchImage;
            p2.SizeMode = PictureBoxSizeMode.StretchImage;
            p3.SizeMode = PictureBoxSizeMode.StretchImage;
            p4.SizeMode = PictureBoxSizeMode.StretchImage;

            // Add locations
            p1.Left = 20;
            p1.Top = 20;
            p2.Left = 110;
            p2.Top = 20;
            p3.Left = 200;
            p3.Top = 20;
            p4.Left = 290;
            p4.Top = 20;



        }
```

Check results



Using other than default events of the control

Adding *mouse* events

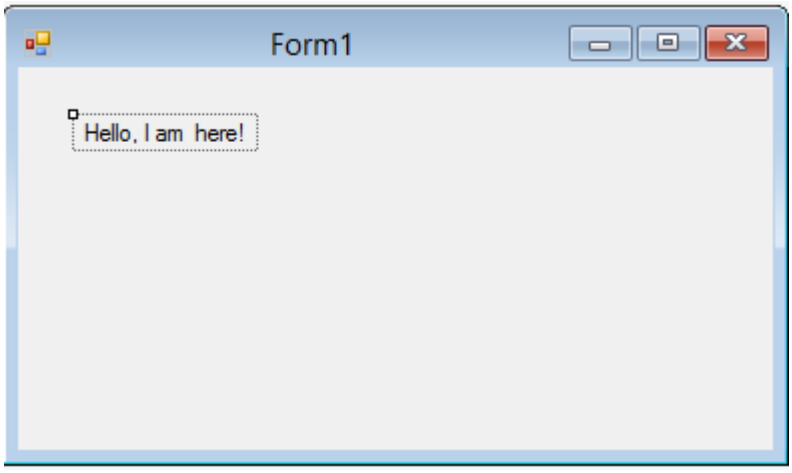
Example:
Mouse events above a label

Create a nexw project

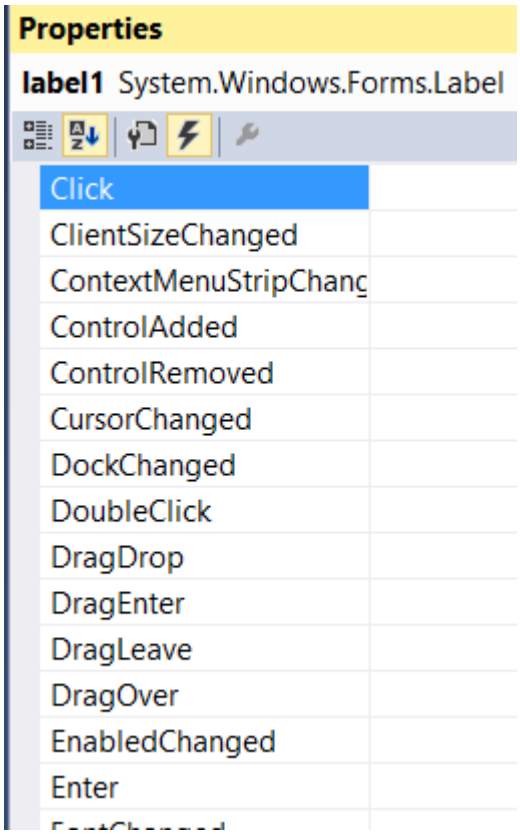
Add there a label

Now:
- when the mouse cursor comes above the label, some text is shown on the form
- when the mouse cursor exits the label, some text is shown on the form
- when the mouse is clicked on the label, some text is shown on the form

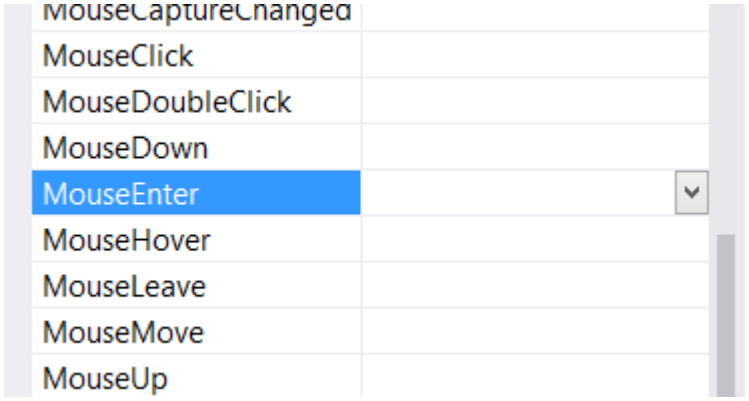
Choose the label



Now from the Properties Box, choose events



Choose MouseEnter event



DoubleClick the box on the right side of the eventname

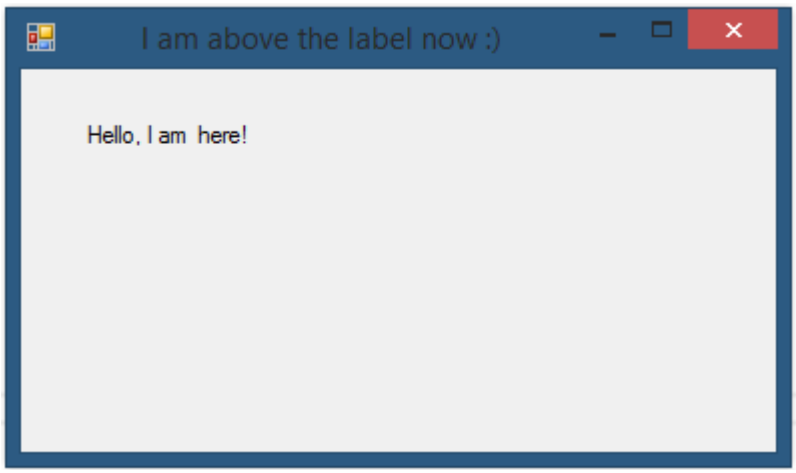
1) Code screen is opened

```
private void label1_MouseEnter(object sender, EventArgs e)
{

}
```

Write the code:

```
private void label1_MouseEnter(object sender, EventArgs e)
{
    this.Text = "I am above the label now :)";
}
```

Test



Try the other mouse events now!

Handling Key Events

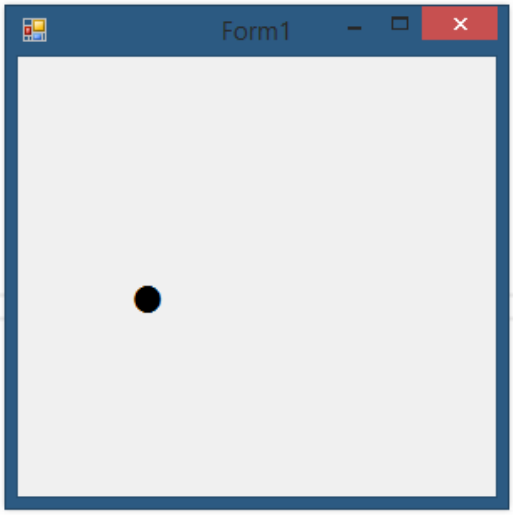
Add a label to the form
Change labels text to show a ball (wingdings and small l)

Add KeyDown event to the form.

Change form's KeyPreview proepty to true.

Do like this:

```
int sx = 0;
int sy = 0;
private void Form1_KeyDown(object sender, KeyEventArgs e)
{
    label1.Left += sx;
    label1.Top += sy;
    if (e.KeyCode == Keys.Left)
    {
        sx = -10;
        sy = 0;
    }

        if (e.KeyCode == Keys.Right)
        {
            sx = 10;
            sy = 0;
        }


        if (e.KeyCode == Keys.Up)
        {
            sx = 0;
            sy = -10;
        }

        if (e.KeyCode == Keys.Down)
        {
            sx = 0;
            sy = 10;
        }
    }
```

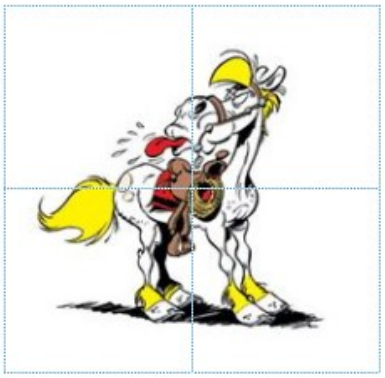


Lets create a puzzle!

Get some picture and split it!

Open this

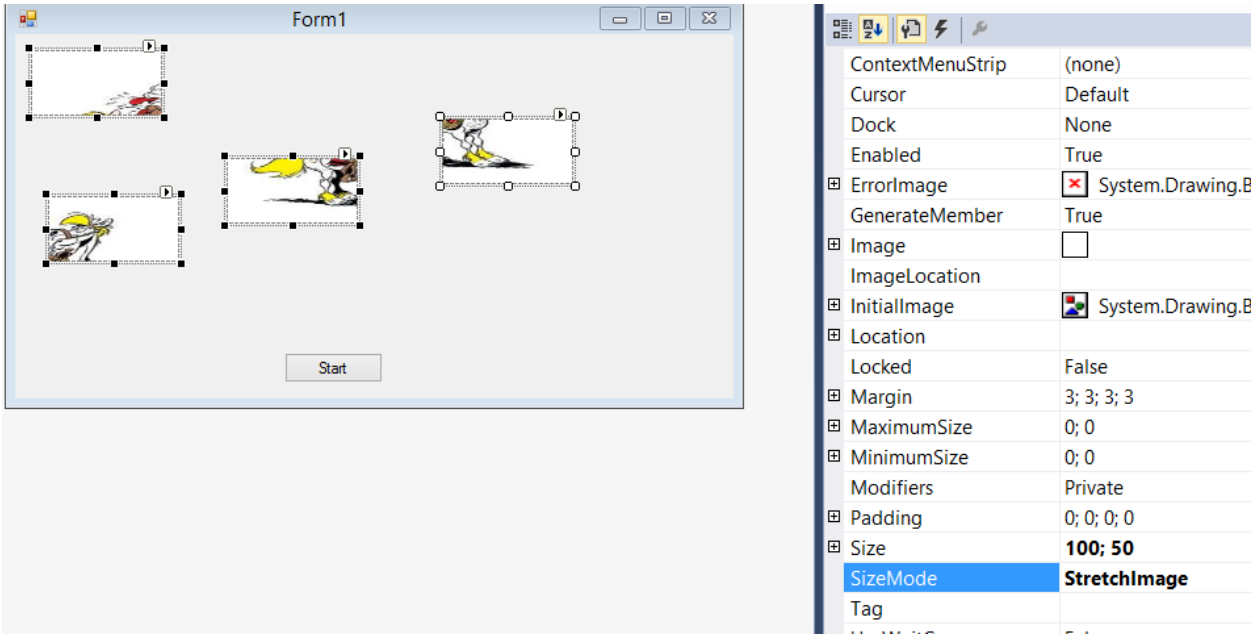
http://imagesplitter.net/



You get this
jj [www.imagesplitter.net].zip

Extract it and add pics to your project.

Add pictureboxes and pics:

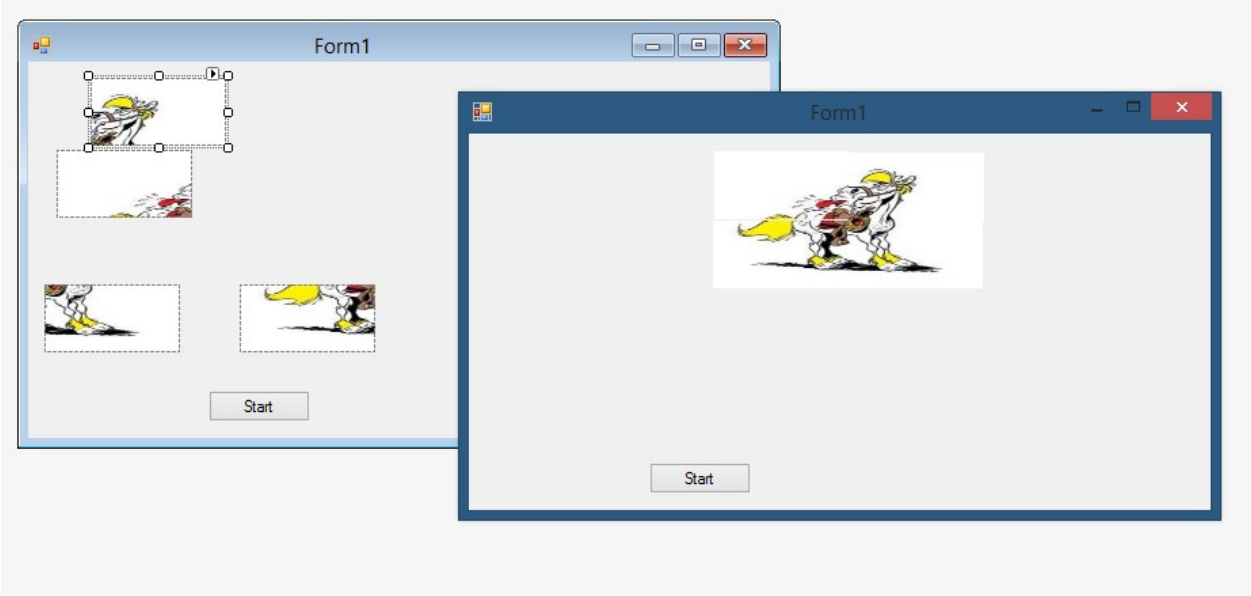


Write the codes for mouse events for every picturebox:

```
int x;
int y;
    private void pictureBox1_MouseDown(object sender, MouseEventArgs
e)
    {
        if (e.Button == MouseButtons.Left) {
        x = e.X;
        y = e.Y;
        }
    }


    private void pictureBox1_MouseMove(object sender, MouseEventArgs
e)
    {
    if (e.Button == MouseButtons.Left) {
            pictureBox1.Left += (e.X - x);
            pictureBox1.Top += (e.Y - y);
```

```
    }
}
```

Test



Here is shown another pic series for the puzzle use:

Animation

Create an animation that presents you or your home town.

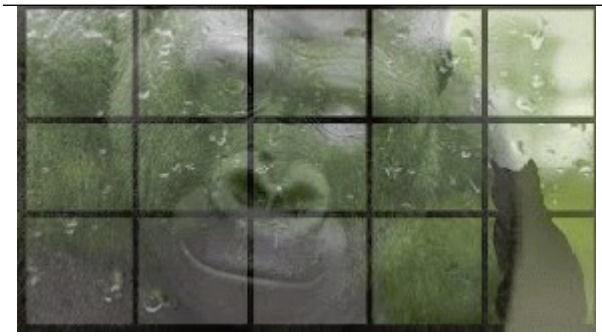
Add some 5 pictures there. Even texts can be added.

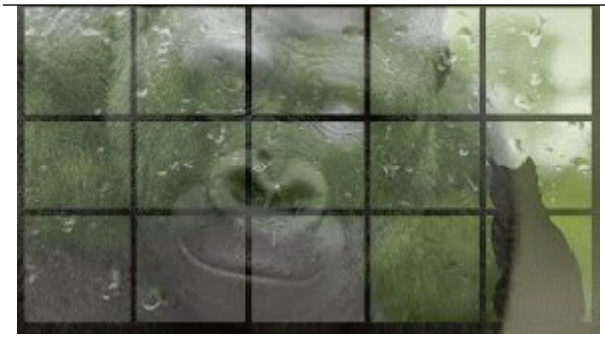
Add picturebox.

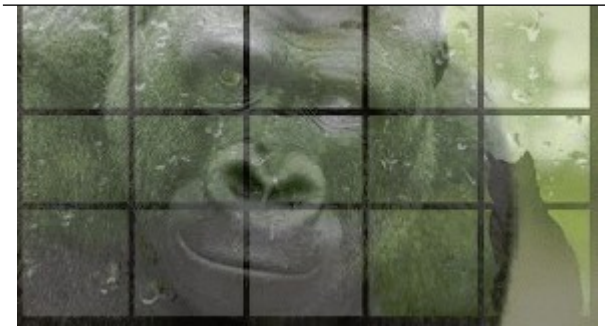
Add timer.

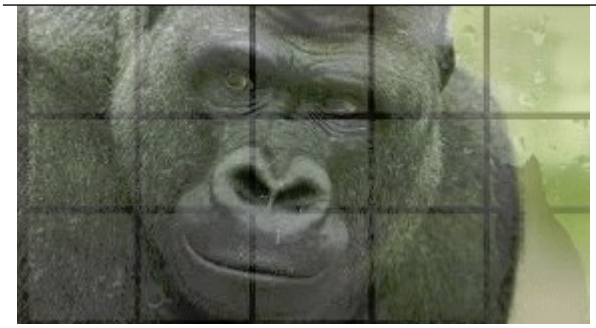
With timer_tick() a new image is added to the picturebox.
Add start and stop buttons.

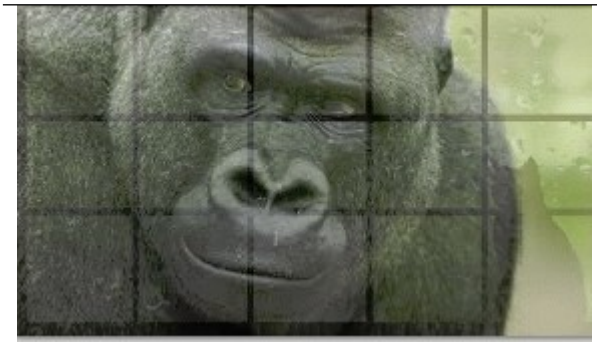
Or do an animation following one of these examples – here all different images are shown:



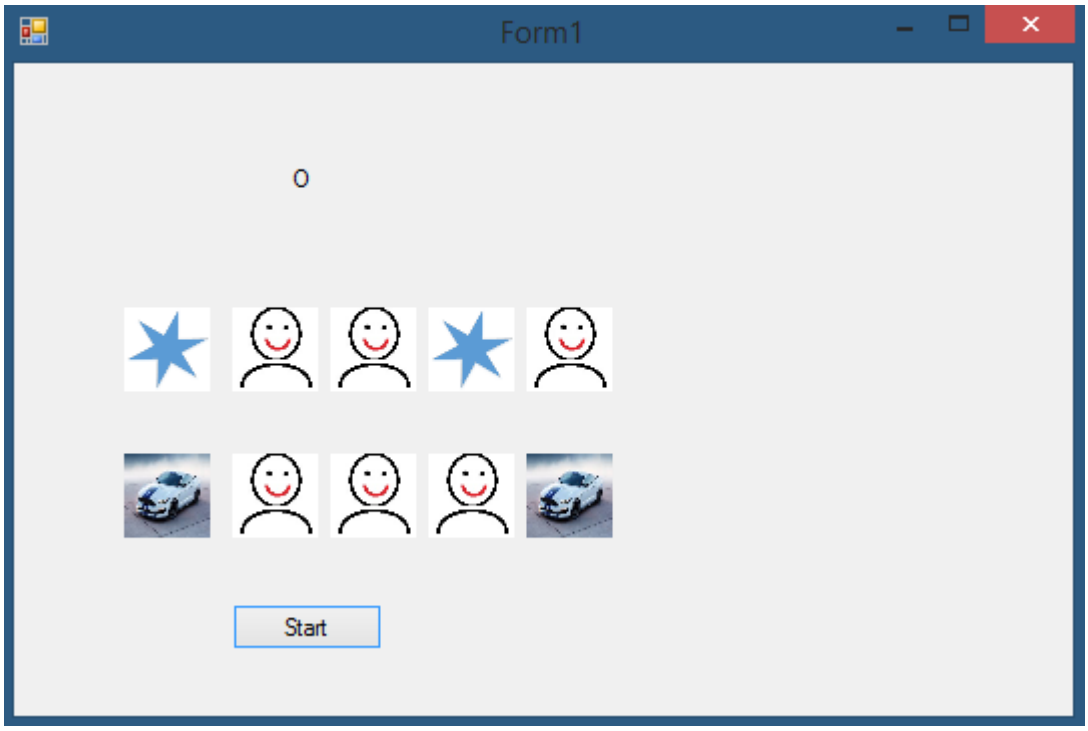




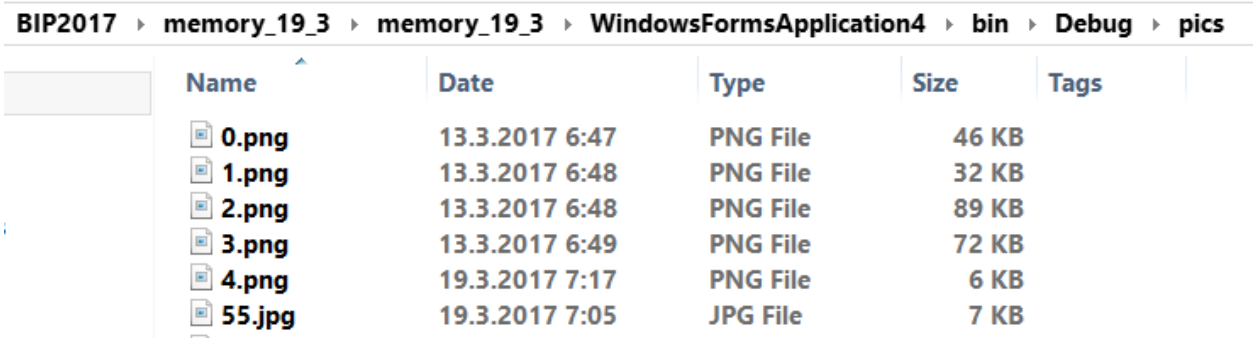




Memory Game

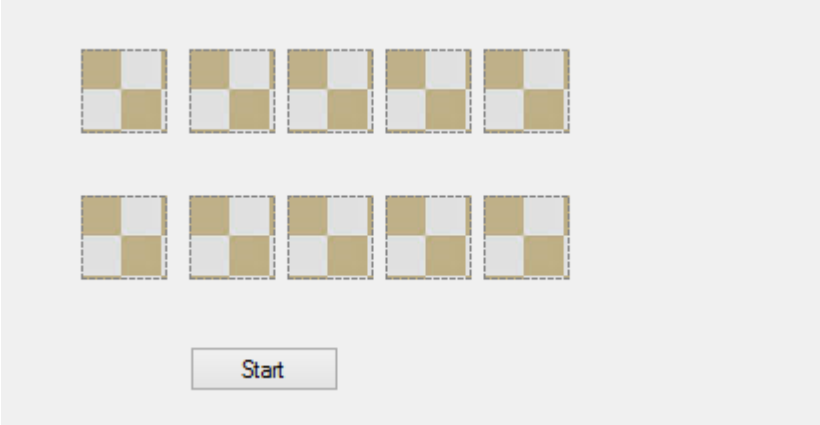


Search first 5 pictures and save them.
Name them like this:



Picture 55.jpg will work as a flipped picture.

Start a new project and add there a button and 10 pictureboxes:



Set all pctureboxes disabled. They are then enabled when you pust Start button.

Put these definitions to your codefile:

```
String filePath = "pics\\";
String[] images = new String[10];
Random rr = new Random();
```

Names of images are added to images array.
Then array is shuffled by using a for loop where we change places of different images.

Put this code behind start button's click event:

```
for (int k = 0; k < 5; k++)
            images[k] = filePath + k + ".png";

        int p = 0;
        for (int k = 5; k < 10; k++)
        {
            images[k] = filePath + p + ".png";
            p++;
        }

        for (int k = 0; k < 1000; k++)
```

```
{
    int a = rr.Next(0, 10);
    int b = rr.Next(0, 10);
    String temp = images[a];
    images[a] = images[b];
    images[b] = temp;

}
```

```
pictureBox1.SizeMode = PictureBoxSizeMode.StretchImage;
pictureBox2.SizeMode = PictureBoxSizeMode.StretchImage;
pictureBox3.SizeMode = PictureBoxSizeMode.StretchImage;
pictureBox4.SizeMode = PictureBoxSizeMode.StretchImage;
pictureBox5.SizeMode = PictureBoxSizeMode.StretchImage;

pictureBox6.SizeMode = PictureBoxSizeMode.StretchImage;
pictureBox7.SizeMode = PictureBoxSizeMode.StretchImage;
pictureBox8.SizeMode = PictureBoxSizeMode.StretchImage;
pictureBox9.SizeMode = PictureBoxSizeMode.StretchImage;
pictureBox10.SizeMode = PictureBoxSizeMode.StretchImage;


// put all pictureboxes enabled
foreach (Control pp in Controls)
{
    pp.Enabled = true;
}
```

Define then these variables:

```
String prevPic = "";
String nextPic = "";
int count = 0;
PictureBox prev, next;
```

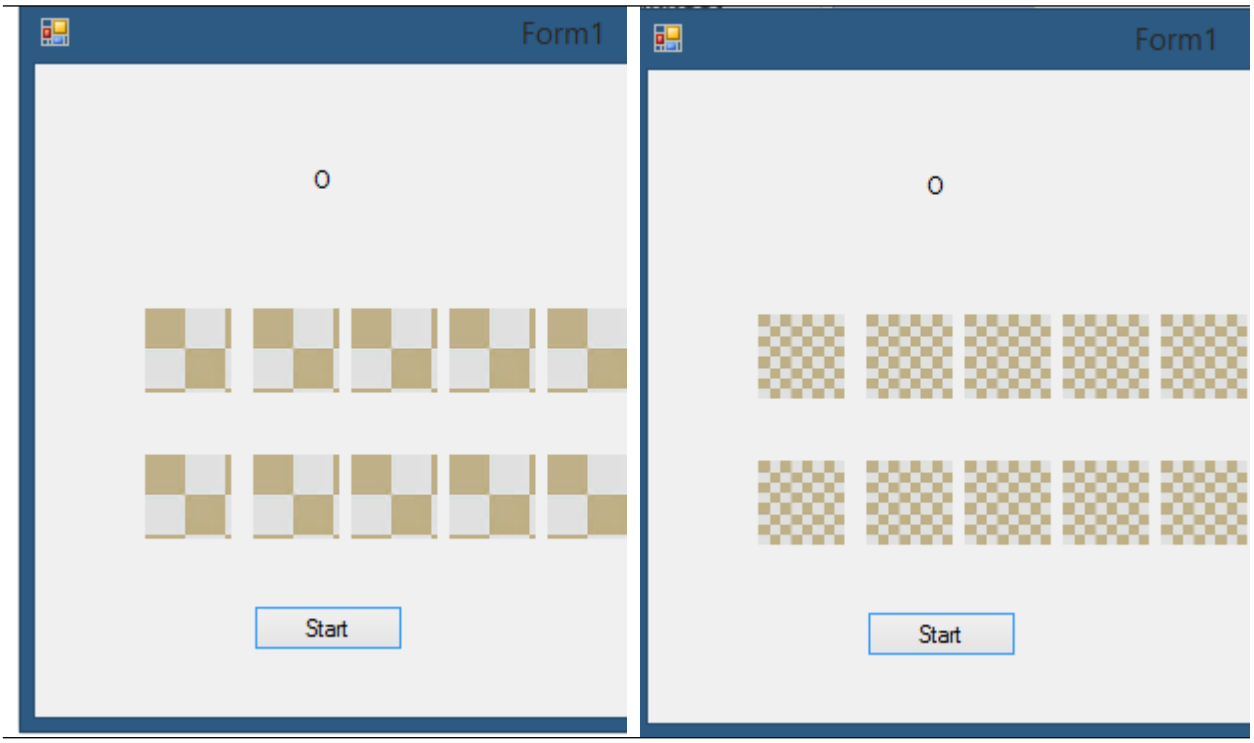
Add this code behind every picturebox'es click event:

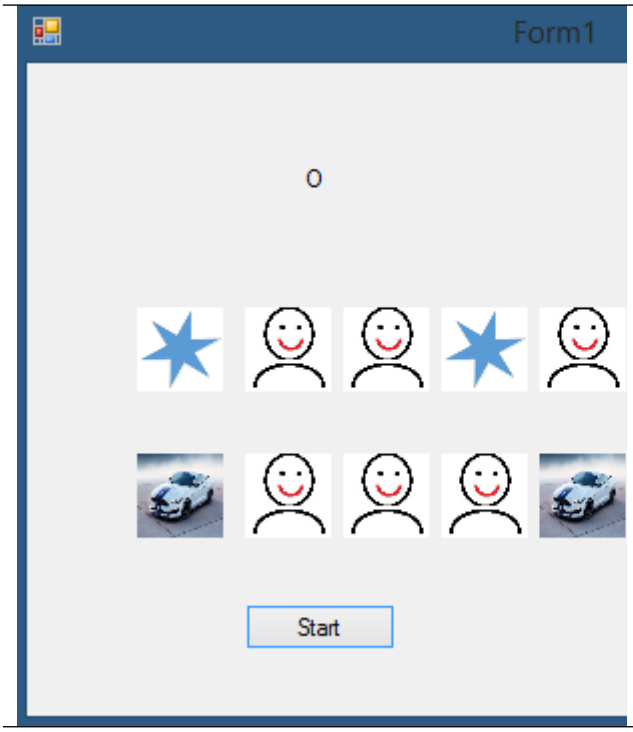
Here we use pictureBox1 (remember to change names when you use this code with other pictureboxes)

```
pictureBox1.Image = Image.FromFile(images[0]);
count++;
String name = pictureBox1.Name;

String nr = name.Substring(name.Length - 1);
int n = Convert.ToUInt16(nr);
n--;
String pic = images[n];

if (count == 1)
{
    prevPic = pic;
    prev = pictureBox1;
}
else
{
    nextPic = pic;
    if (prevPic.Equals(nextPic))
    {
```

```
                prev.Enabled = false;
                pictureBox1.Enabled = false;
            }
            else
            {
                prev.Image = Image.FromFile(filePath + "55.jpg");
                pictureBox1.Image = Image.FromFile(filePath +
               "55.jpg");
            }
            count = 0;
        }


    }
```

Try you app!

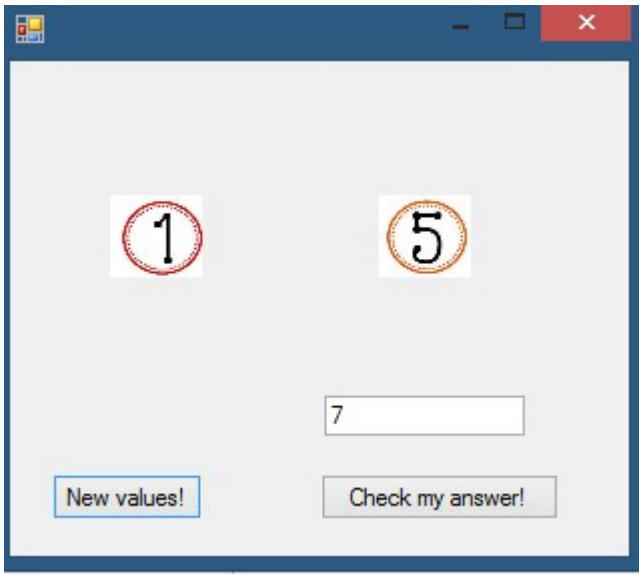




Try to add there new features!

Falling Numbers

Let's create a Falling Numbers Math Game



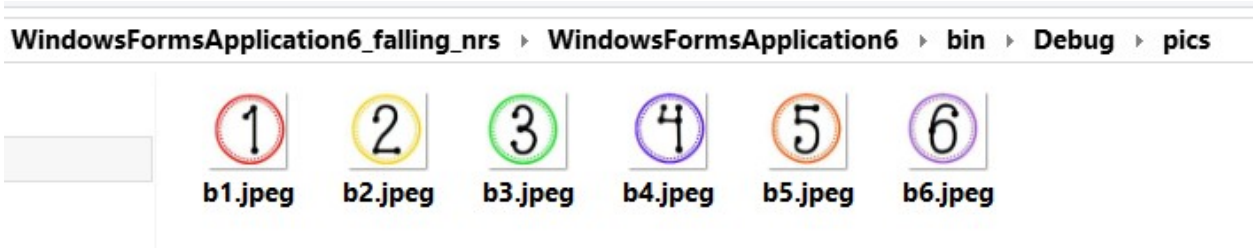
With "New values!" button new numbers are generated and they are moving downwards.

User types the sum of values to the textbox and it is checked by clicking the

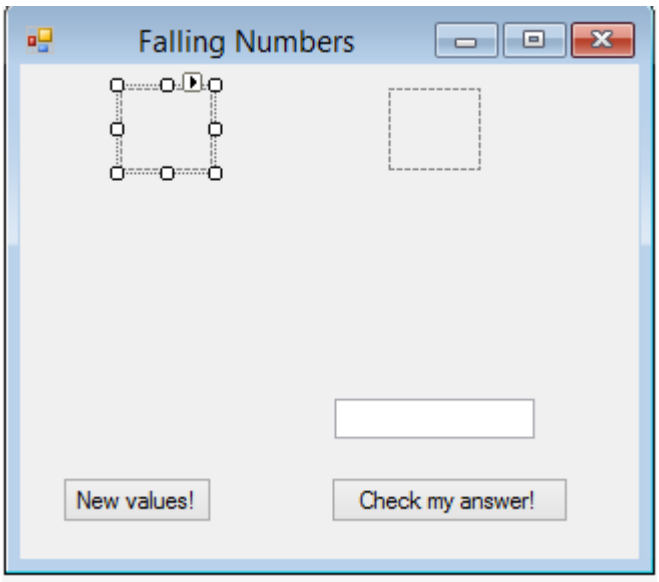
"Check my answer!" –button.

Answer has to be given and checked before falling numbers reach the foot of the form!

Add first number images to your project:



Add 2 pictureboxes, one textbox and 2 buttons to your form:



Add timer to your project.

Write this code behind buttons 1 click event:

```
Random rr = new Random();
```

```
        String path = "pics\\";
        int pic1, pic2;
        private void button1_Click(object sender, EventArgs e)
        {
            this.Text = "";

            pic1 = rr.Next(1, 7);
            pic2 = rr.Next(1, 7);
            pictureBox1.Top = 12;
            pictureBox2.Top = 12;
            switch (pic1)
                {
                 case 1: pictureBox1.Image = Image.FromFile(path +
"b1.jpeg"); break;
                 case 2: pictureBox1.Image = Image.FromFile(path +
"b2.jpeg"); break;
                case 3: pictureBox1.Image = Image.FromFile(path +
"b3.jpeg"); break;
                case 4: pictureBox1.Image = Image.FromFile(path +
"b4.jpeg"); break;
                case 5: pictureBox1.Image = Image.FromFile(path +
"b5.jpeg"); break;
                case 6: pictureBox1.Image = Image.FromFile(path +
"b6.jpeg"); break;

            }

            switch (pic2)
            {
                case 1: pictureBox2.Image = Image.FromFile(path +
"b1.jpeg"); break;
                case 2: pictureBox2.Image = Image.FromFile(path +
"b2.jpeg"); break;
                case 3: pictureBox2.Image = Image.FromFile(path +
"b3.jpeg"); break;
```

```
                case 4: pictureBox2.Image = Image.FromFile(path +
"b4.jpeg"); break;
                case 5: pictureBox2.Image = Image.FromFile(path +
"b5.jpeg"); break;
                case 6: pictureBox2.Image = Image.FromFile(path +
"b6.jpeg"); break;
            }



            pictureBox1.Visible = true;
            pictureBox2.Visible = true;

            timer1.Enabled = true;
            correct = pic1 + pic2;
        }¨
```

Write this code behind button's 2 click event:

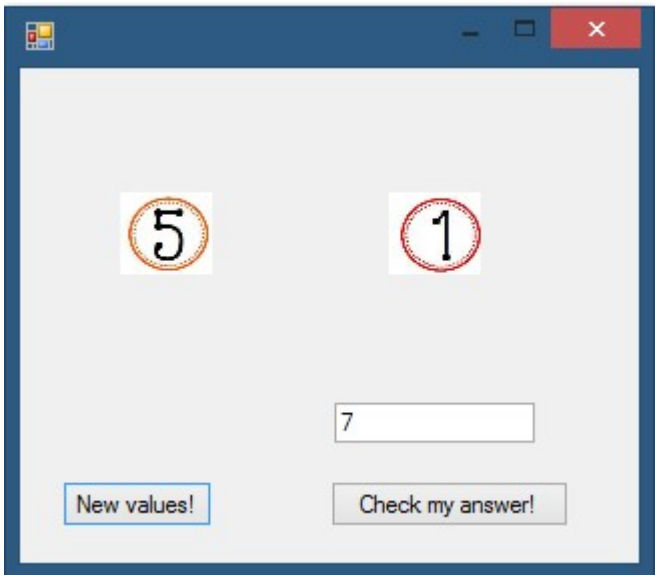
```
 int answer;
        int correct;

        private void button2_Click(object sender, EventArgs e)
        {
            answer = Convert.ToInt16(textBox1.Text);
            if (answer == correct)
            {
                timer1.Enabled = false;
                this.Text = "JEEEE";
            }
            else
                this.Text = "WRONG!!!!!!";
```

```
}
```

Write this code behind timer's TimerTick event:

```
int step = 5;
 private void timer1_Tick(object sender, EventArgs e)
 {
     pictureBox1.Top += step;
     pictureBox2.Top += step;
 }
```

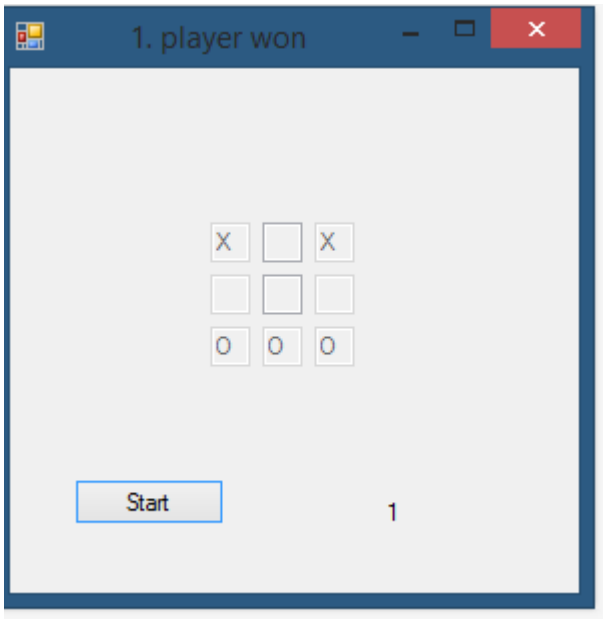
Try it!



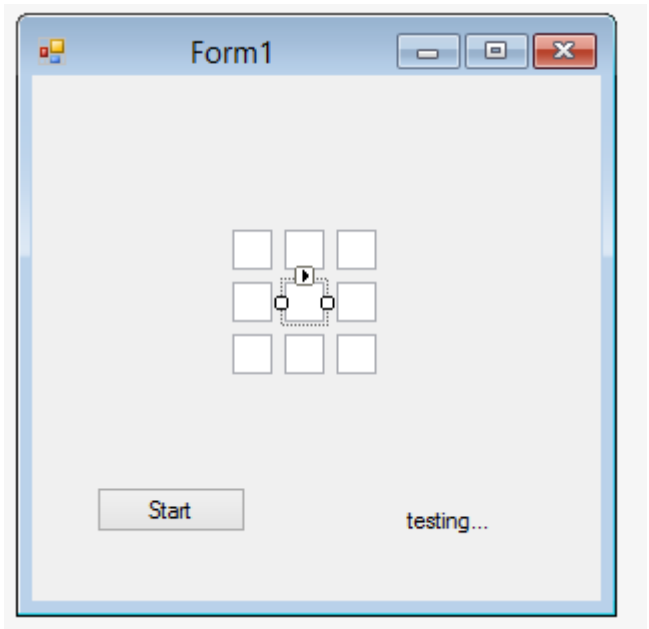
Try to put there more e.g. rewarding features...

TicTacToe

Let's create a TicTacToe!



Add 9 textboxes, button and a label to your form:



Define first these variables and arrays:

```
int[,] r = new int[3, 3];
int user = 0;
string sign = "O";
TextBox[,] boxes = new TextBox[3,3];
```

Add this code to handle button's click event:

```
private void button1_Click(object sender, EventArgs e)
```

```
        {
            if (user == 1)
                sign = "X";
            for (int i = 0; i < 3; i++)
                for (int j = 0; j < 3; j++)
                    r[i, j] = 0;
            boxes[0, 0] = textBox1;
            boxes[0, 1] = textBox2;
            boxes[0, 2] = textBox3;

            boxes[1, 0] = textBox4;
            boxes[1, 1] = textBox5;
            boxes[1, 2] = textBox6;

            boxes[2, 0] = textBox7;
            boxes[2, 1] = textBox8;
            boxes[2, 2] = textBox9;

        }
```

Add this code behind every textbox's click event:

```
private void textBox1_Click(object sender, EventArgs e)
        {
            if (textBox1.Text.Length == 0)
            {
                textBox1.Text = sign;
                textBox1.Enabled = false;
            }

            if (sign.Equals("O"))
                sign = "X";
```

```
            else
                sign = "O";

            label1.Text = "" + check();
        }
```

Create this code to check the game situation after every move:

```
public int check()
        {
           int winner = 0;
           for (int i = 0; i < 3; i++)
           {
               if (boxes[i, 0].Text == boxes[i, 1].Text && boxes[i,
0].Text == boxes[i, 2].Text
                   )
                   if (boxes[i, 0].Text == "O")
                   { winner = 1; break; }
                   else if (boxes[i, 0].Text == "X")
                   { winner = 2; break; }
           }

           for (int i = 0; i < 3; i++)
           {
               if (boxes[0, i].Text == boxes[1, i].Text && boxes[0,
i].Text == boxes[2, i].Text
                   )
                   if (boxes[0, i].Text == "O")
                   { winner = 1; break; }
                   else if (boxes[0, i].Text == "X")
                   { winner = 2; break; }
           }
```

```
            if (boxes[0, 0].Text == boxes[1, 1].Text && boxes[0, 0].Text
== boxes[2, 2].Text)
            {
                if (boxes[0, 0].Text == "O")
                { winner = 1; }
                else if (boxes[0, 0].Text == "X")
                { winner = 2; }
            }



            if (boxes[0, 2].Text == boxes[1, 1].Text && boxes[0, 2].Text
== boxes[2, 0].Text)
            {
                if (boxes[0, 2].Text == "O")
                { winner = 1; }
                else if (boxes[0, 2].Text == "X")
                { winner = 2; }
            }


             if (winner == 1)
                 this.Text = "1. player won";
             if (winner == 2)
                 this.Text = "2. player won";


             if (winner == 1 || winner == 2)
             {


                 for (int i = 0; i < 3; i++)
                     for (int j = 0; j < 3; j++)
                         boxes[i,j].Enabled = false;
                 if (winner == 1)
                     label1.Text = "you won!";
```

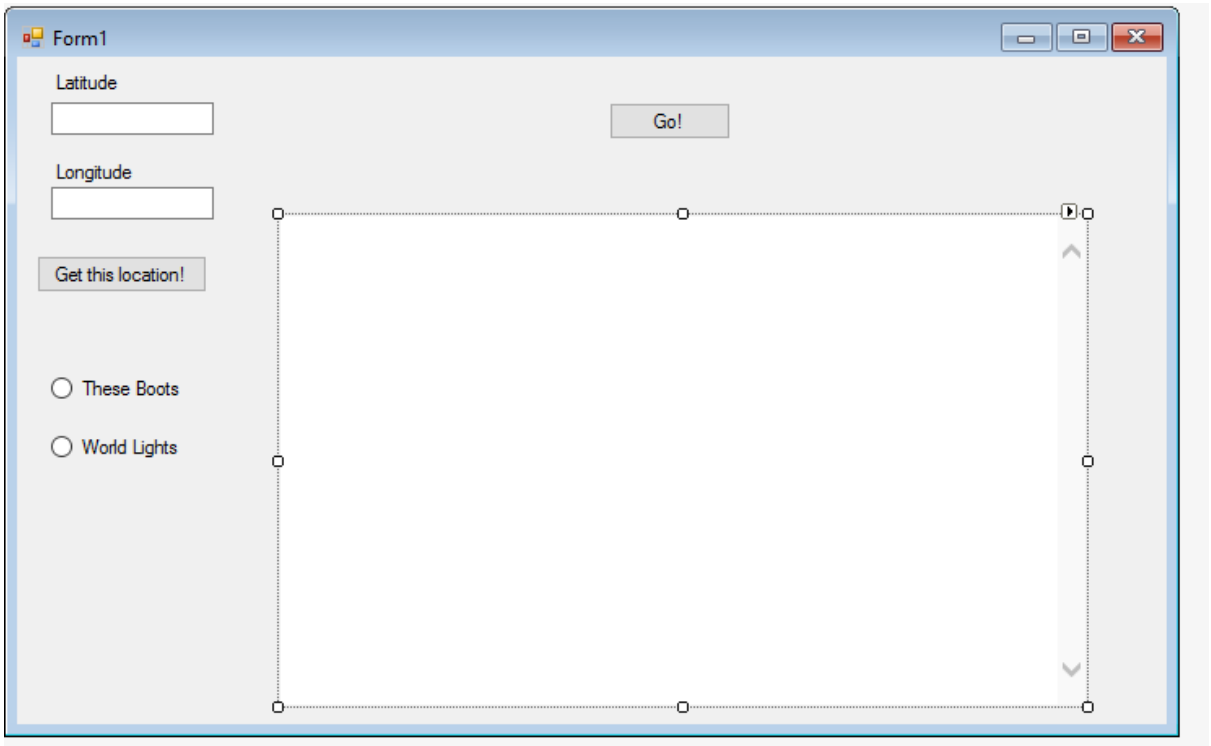
```
            else
                label1.Text = "PC won!";

        }
        return winner;

    }
```

Create own web browser

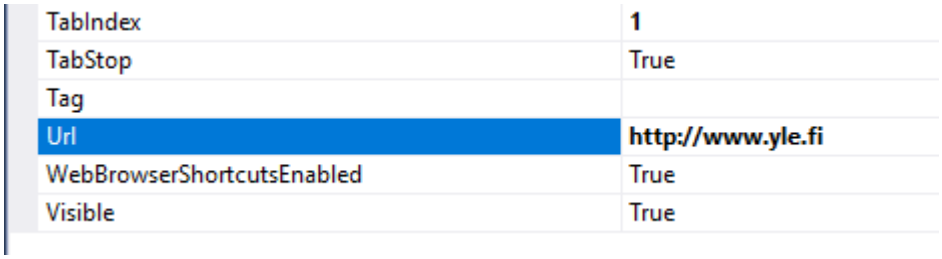
Let's create a web browser

Create this kind of gui: 2 buttons, 2 textboxes, one webbrowser.

(Radionbuttons are needed for the next stage)



Web browser has an address:



TabIndex	1
TabStop	True
Tag	
Url	http://www.yle.fi
WebBrowserShortcutsEnabled	True
Visible	True

When user clicks the Go! button, shows web browser given Internet page:

```
    private void button1_Click(object sender, EventArgs e)
        {
            webBrowser1.Show();
        }
```

Then user can write coordinates (latitude and longitude) to textboxes. With button Get this location a google map url is created with given coordinates and that page is shown by web browser:

```
    private void button2_Click(object sender, EventArgs e)
        {
            String lat = textBox1.Text;
            String lon = textBox2.Text;

            String addr = @"https://www.google.fi/maps/@" +
            lat + "," + lon;

          Uri uri = new Uri(addr);
           webBrowser1.Url = uri;

        }
```

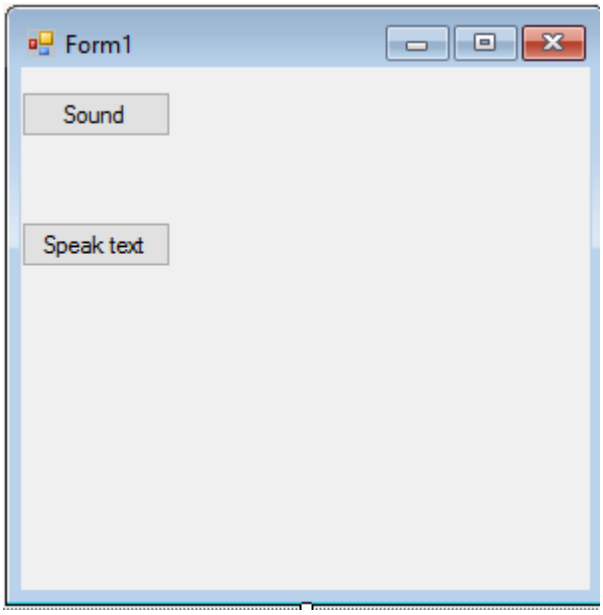
Try it!

With radiobuttons we can add youtube music videos to be shown by web browser:

```
    private void radioButton1_CheckedChanged(object sender, EventArgs e)
        {
            //https://www.youtube.com/watch?v=SbyAZQ45uww

            String addr = @"https://www.youtube.com/watch?v=SbyAZQ45uww";

          Uri uri = new Uri(addr);
          webBrowser1.Url = uri;

        }

        private void radioButton2_CheckedChanged(object sender,
                                            EventArgs e)
        {
            String addr = @"https://www.youtube.com/watch?v=JW4LFHjU15Q";

          Uri uri = new Uri(addr);
          webBrowser1.Url = uri;
        }
```

Sounds

Create this gui



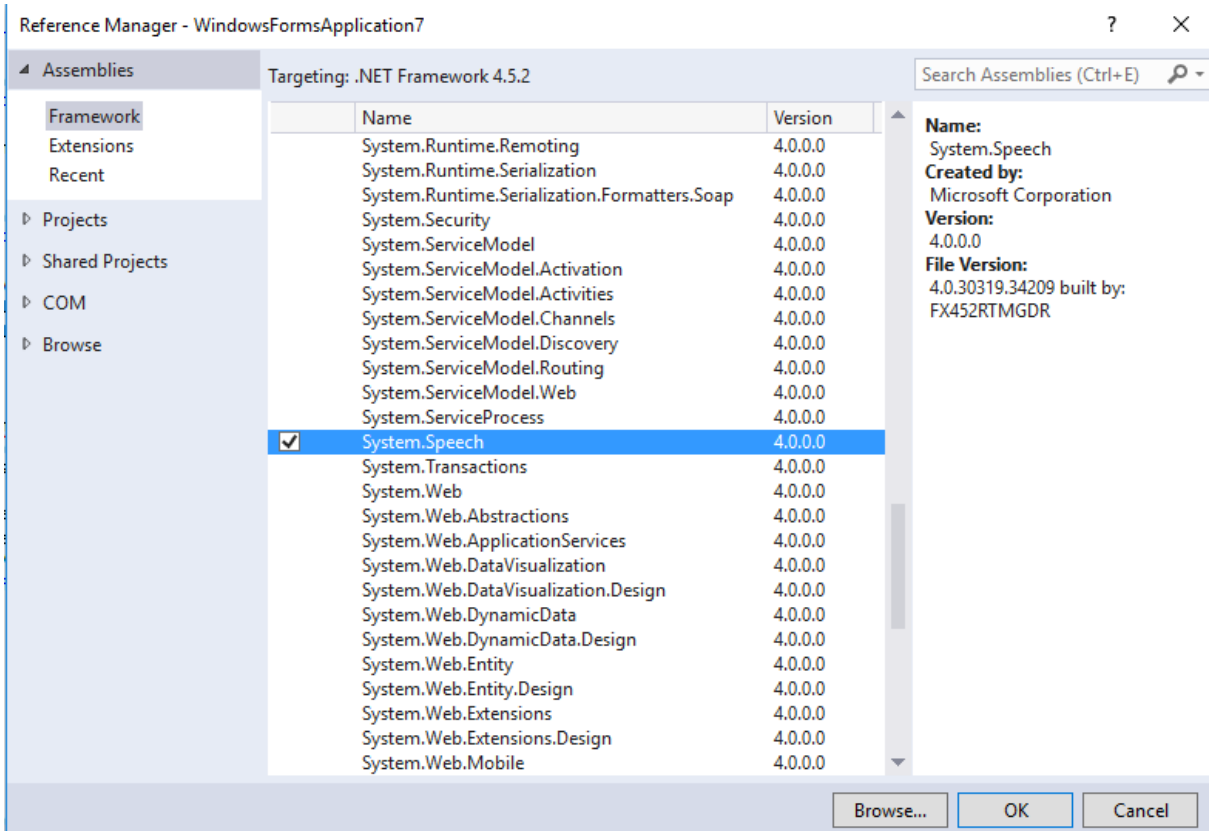
With sound you can listen sounds:

```
private void button1_Click(object sender, EventArgs e)
        {
            System.Media.SoundPlayer player;
            player = new System.Media.SoundPlayer(@"pig3.wav");
            player.Play();
        }
```

Speech

Now you can listen texts!

Add reference



And add these

```
using System.Speech;
using System.Speech.Synthesis;
```

Add this code:

```
SpeechSynthesizer reader;
void sound(String text)
{
    reader = new SpeechSynthesizer();
    if (text != "")    //if text area is not empty
    {
        reader.SpeakAsync(text);
    }
}

private void button2_Click(object sender, EventArgs e)
```

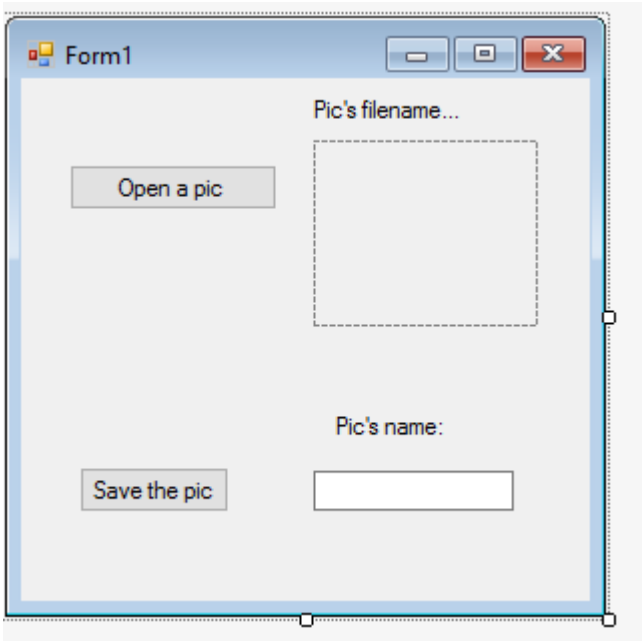
```
{
    sound("hello, all");
}
```

Try it!

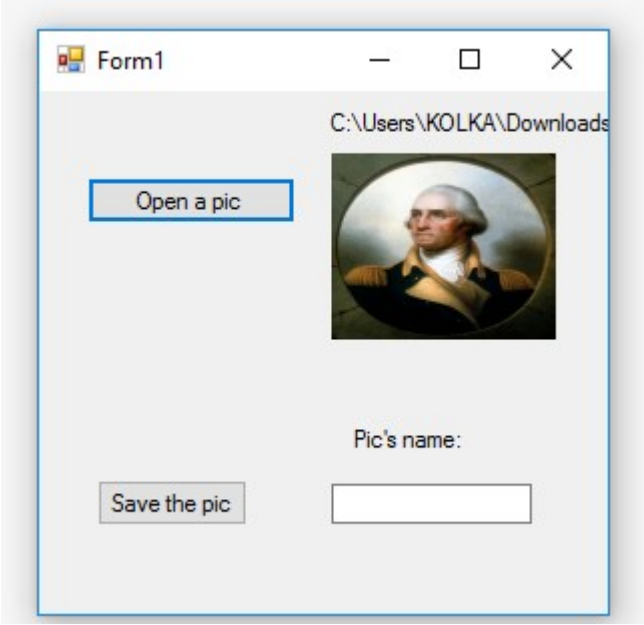
File handling

Opening a file and saving it

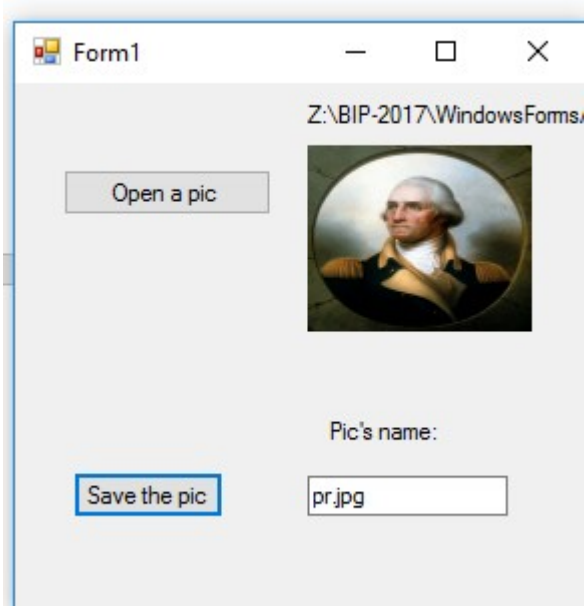
Create this gui:

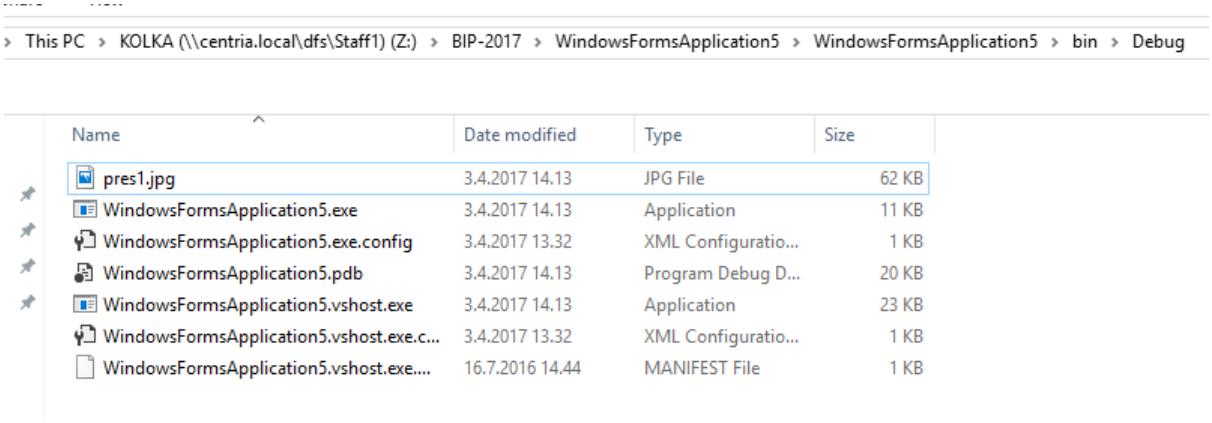


With Open a pic a file opening screen is opening and you can chooce a pic.



With Save the pic pic's filename is taken from textbox and the file is saved to debug-folder.



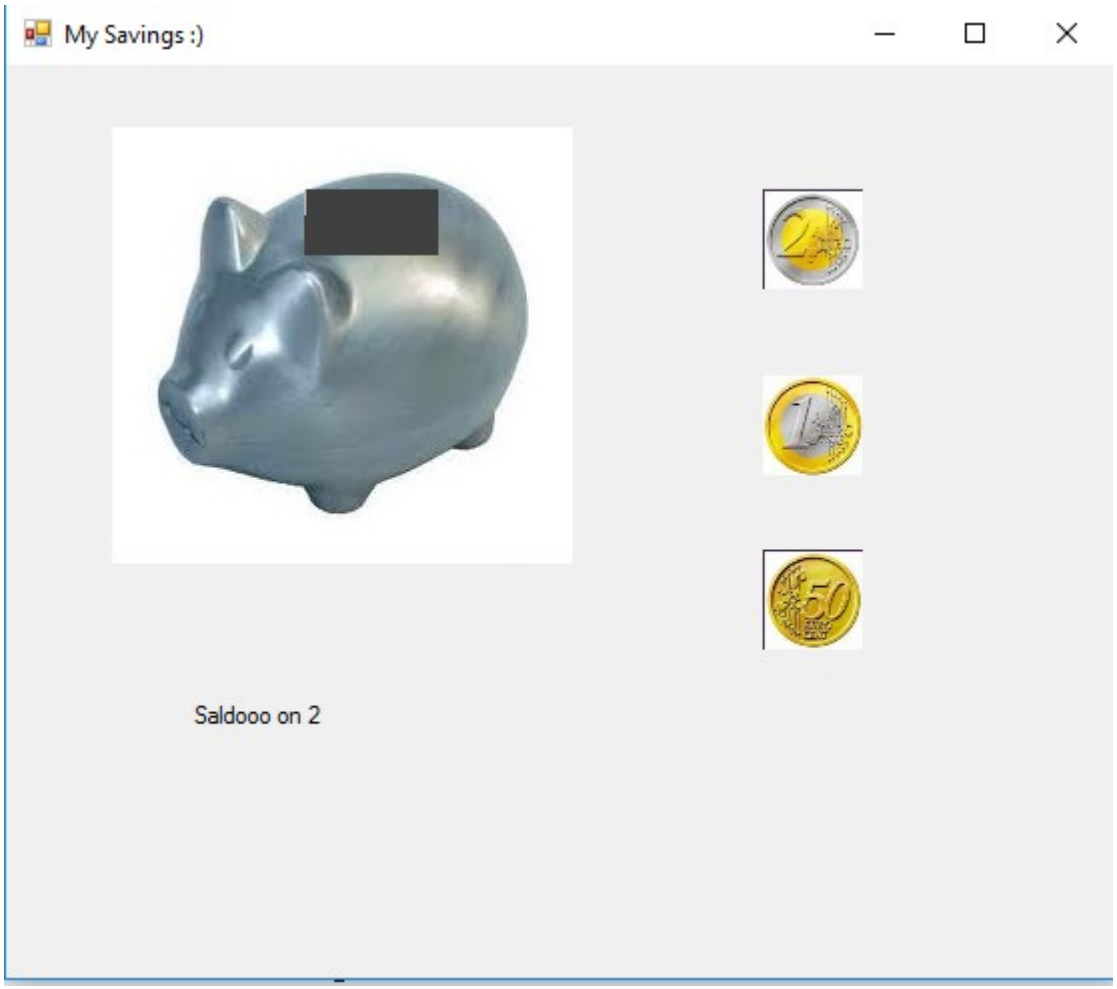


Codes

```
private void button1_Click(object sender, EventArgs e)
  {
      OpenFileDialog open = new OpenFileDialog();
      // image filters
      open.Filter = "Image Files(*.jpg; *.jpeg; *.gif;
       *.bmp)|*.jpg;  *.jpeg; *.gif; *.bmp";
      if (open.ShowDialog() == DialogResult.OK)
      {
          // display image in picture box
          pictureBox1.Image = new Bitmap(open.FileName);
          // image file path
          label1.Text = open.FileName;
      }
  }

  private void button2_Click(object sender, EventArgs e)
  {
      string FilePath = Application.StartupPath + "\\" +
      textBox1.Text;
      label1.Text = "" + FilePath;
      pictureBox1.Image.Save(FilePath);
}
```

Small project: Drag and drop

Create this gui:



Add these codes:

```
double summa = 0;
  private void pictureBox1_DragDrop(object sender, DragEventArgs e)
  {

  }
  int a = 0;

  private void pictureBox2_MouseDown(object sender, MouseEventArgs
e)
```

```
        {
            a = 1;
            pictureBox2.DoDragDrop(this, DragDropEffects.All);
         }




        private void pictureBox3_MouseDown(object sender, MouseEventArgs
e)
        {
            a = 2;
            pictureBox3.DoDragDrop(this, DragDropEffects.All);



        }

        private void pictureBox4_MouseDown(object sender, MouseEventArgs
e)
        {
            a = 3;
            pictureBox4.DoDragDrop(this, DragDropEffects.All);

        }

        private void textBox1_DragEnter(object sender, DragEventArgs e)
        {
            AllowDrop = true;
            e.Effect = DragDropEffects.All;

        }

        private void textBox1_DragDrop(object sender, DragEventArgs e)
```

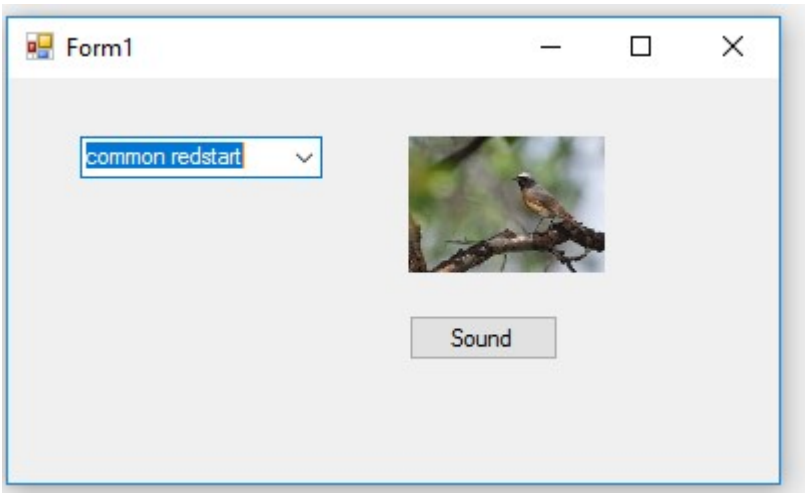
```
{
        AllowDrop = true;
        e.Effect = DragDropEffects.All;

        if (a == 1)
        {
            summa = summa + 2;
            label1.Text = "Balance is " + summa;
        }
        if (a == 2)
        {
            summa = summa + 1;
            label1.Text = "Balance is " + summa;
        }
        if (a == 3)
        {
            summa = summa + 0.50;
            label1.Text = "Balance is " + summa;
    }
  }
```

Try it!

Now, create your own Drag and Drop App!!

Bird info system- exercise

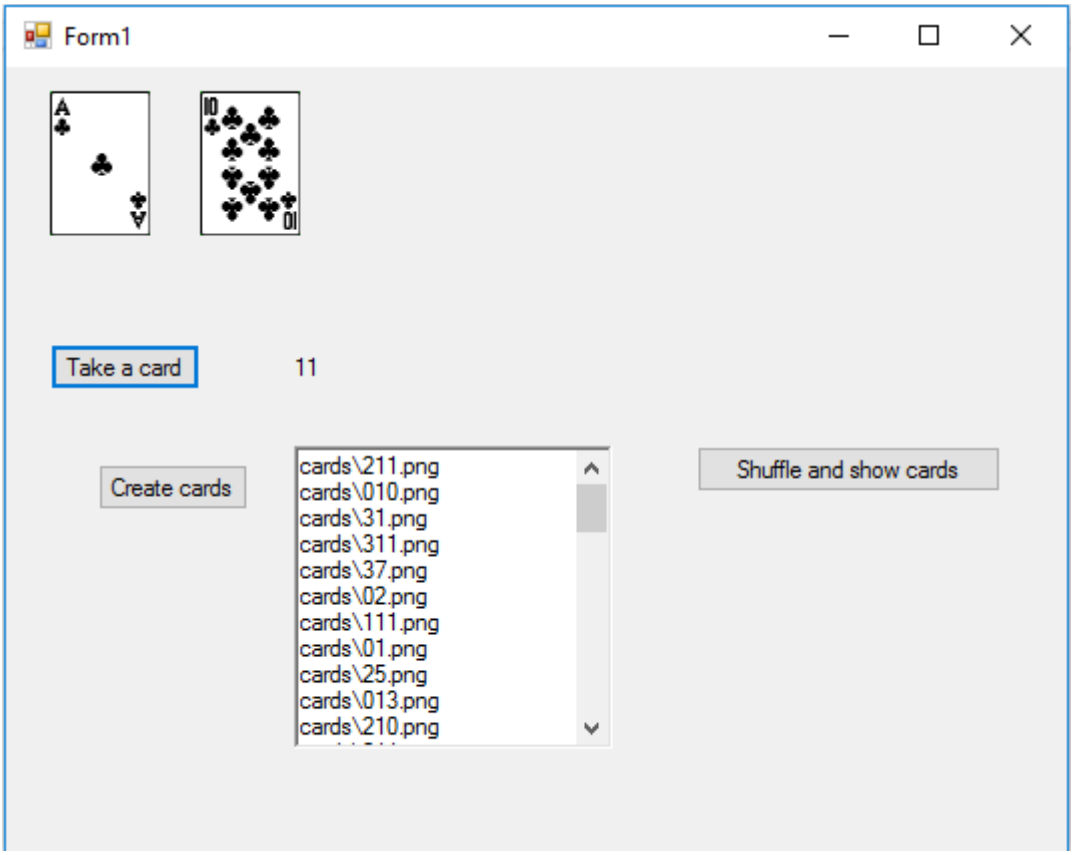
A birdname can be chosen from the list.
Then shows the birds picture and sound.



Simple BlackJack

There is 1 player that tries to get 21 by taking cards one by one from the deck.

Card images are shown on a form.



Codes:

```
        String[] cardFileNames =
          new String[52];

          int amount = 0;
          int points = 0;
          private void button1_Click(object sender, EventArgs e)
          {
              if (amount == 0)
              {
                  pictureBox1.Image =
Image.FromFile(cardFileNames[amount]);

                  String card = cardFileNames[amount];
                  int pointPlace = card.IndexOf('.');

                  String pointPart = card.Substring(7, pointPlace - 7);
                  points += Convert.ToInt16(pointPart);
                  label1.Text = "" + points;
              }
              if (amount == 1)
              {
                  pictureBox2.Image =
Image.FromFile(cardFileNames[amount]);

                  String card = cardFileNames[amount];
                  int pointPlace = card.IndexOf('.');

                  String pointPart = card.Substring(7, pointPlace - 7);
                  points += Convert.ToInt16(pointPart);
                  label1.Text = "" + points;
              }

              if (amount == 2)
```

```
            {
                pictureBox3.Image =
Image.FromFile(cardFileNames[amount]);

                String card = cardFileNames[amount];
                int pointPlace = card.IndexOf('.');

                String pointPart = card.Substring(7, pointPlace - 7);
                points += Convert.ToInt16(pointPart);
                label1.Text = "" + points;
            }

            if (amount == 3)
            {
                pictureBox4.Image =
Image.FromFile(cardFileNames[amount]);

                String card = cardFileNames[amount];
                int pointPlace = card.IndexOf('.');

                String pointPart = card.Substring(7, pointPlace - 7);
                points += Convert.ToInt16(pointPart);
                label1.Text = "" + points;
            }

            if (amount == 4)
            {
                pictureBox5.Image =
Image.FromFile(cardFileNames[amount]);

                String card = cardFileNames[amount];
                int pointPlace = card.IndexOf('.');
```

```
        String pointPart = card.Substring(7, pointPlace - 7);
        points += Convert.ToInt16(pointPart);
        label1.Text = "" + points;
    }



    amount++;


}

private void Form1_Load(object sender, EventArgs e)
{
    int p = 0;
    String name = "";
    for (int c = 0; c < 4; c++)
        for (int r = 1; r < 14; r++)
        {
            name = "cards\\" + c + "" + r + ".png";
            cardFileNames[p] = name;
            p++;
        }
}

private void button2_Click(object sender, EventArgs e)
{
    for (int c = 0; c < 52; c++)
        richTextBox1.AppendText(cardFileNames[c] + "\n");


}
```

```
  Random rr = new Random();
  private void button3_Click(object sender, EventArgs e)
  {
      for (int c = 0; c < 100; c++)
      {
          int a = rr.Next(0, 52);
          int b = rr.Next(0, 52);
          String temp = cardFileNames[a];
          cardFileNames[a] = cardFileNames[b];
          cardFileNames[b] = temp;
      }

      richTextBox1.Clear();
      for (int c = 0; c < 52; c++)
          richTextBox1.AppendText(cardFileNames[c] + "\n");

}
```

Training tasks

1.

World population has been given in an array here

```
long[,] pops = {
{ 1750,600000000},
 {1800,900000000},
 {1850,1200000000},
 {1900,1600000000},
 {1950,2200000000 },
 {1960,3000000000 },
 {1970,3800000000 },
 {1980,4500000000 },
 {1990,5500000000 },
 {2000,6100000000 },
 {2010,6800000000}};
```

Create the array. Print it nicely to a richtextbox. Write the code that finds out what was the population in that specific year. Year is given in a textbox.

2.

Create a bank app.

User can add money and take money and check the balance.

If user tries to take too much money, it is forbidden and a message

is given.

If user tries to take money 3 times without adding any money between

withdrawals the whole account is frozen so that it cannot be used at all, all buttons are disabled and

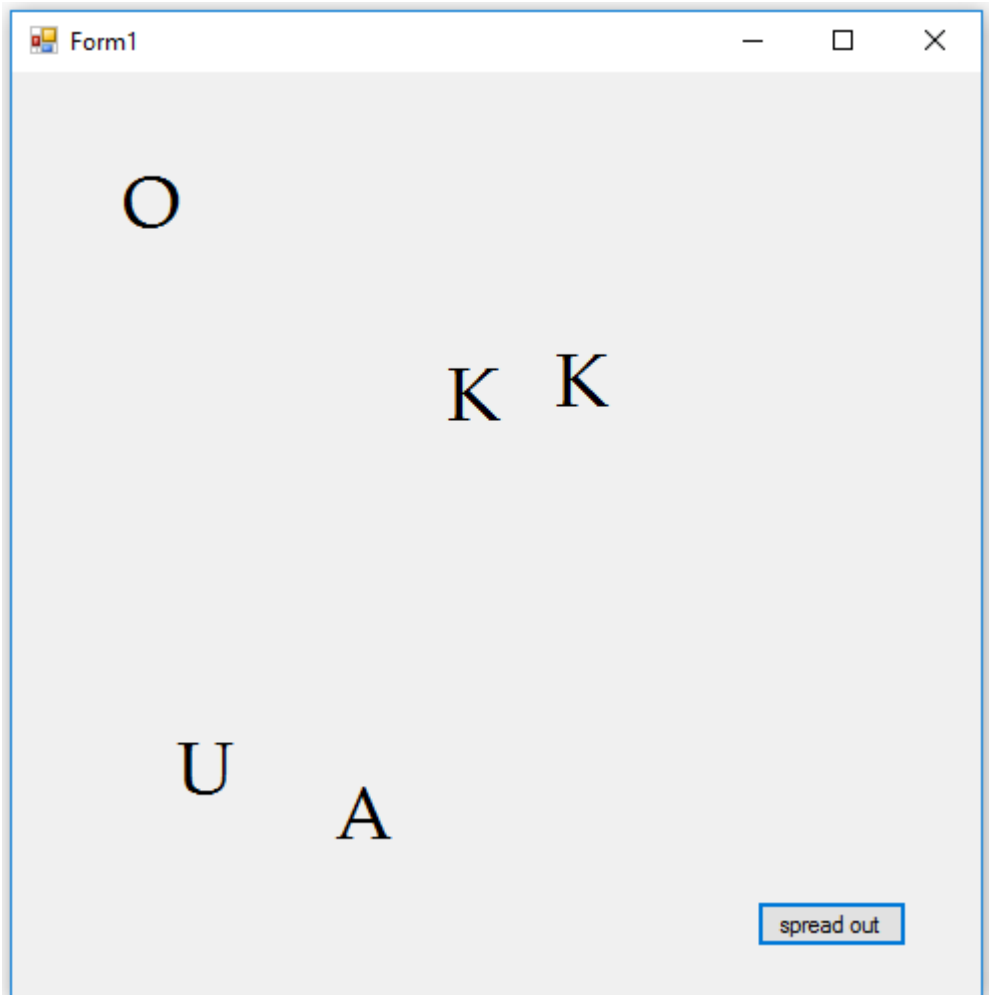
thus, you can not do any transactions!

Only a message of that situation is printed and your name appears to the form (forms' caption).

3.

Create this GUI:

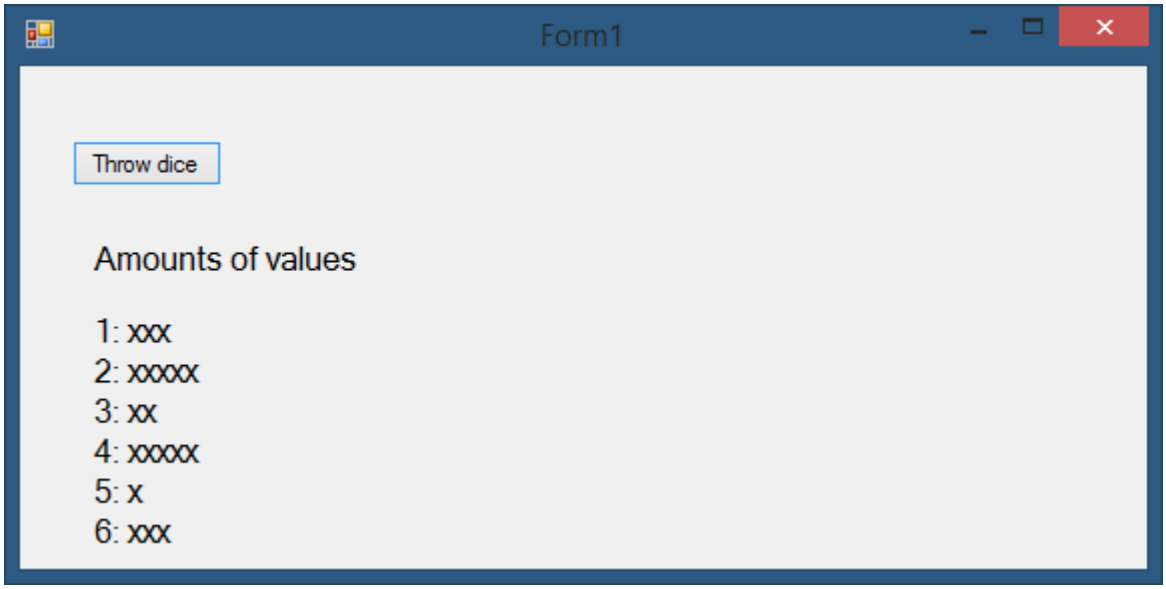
When user clicks the button, characters are spread out to different, random places on the form.



When user then clicks the character, it is moved to the right place again.

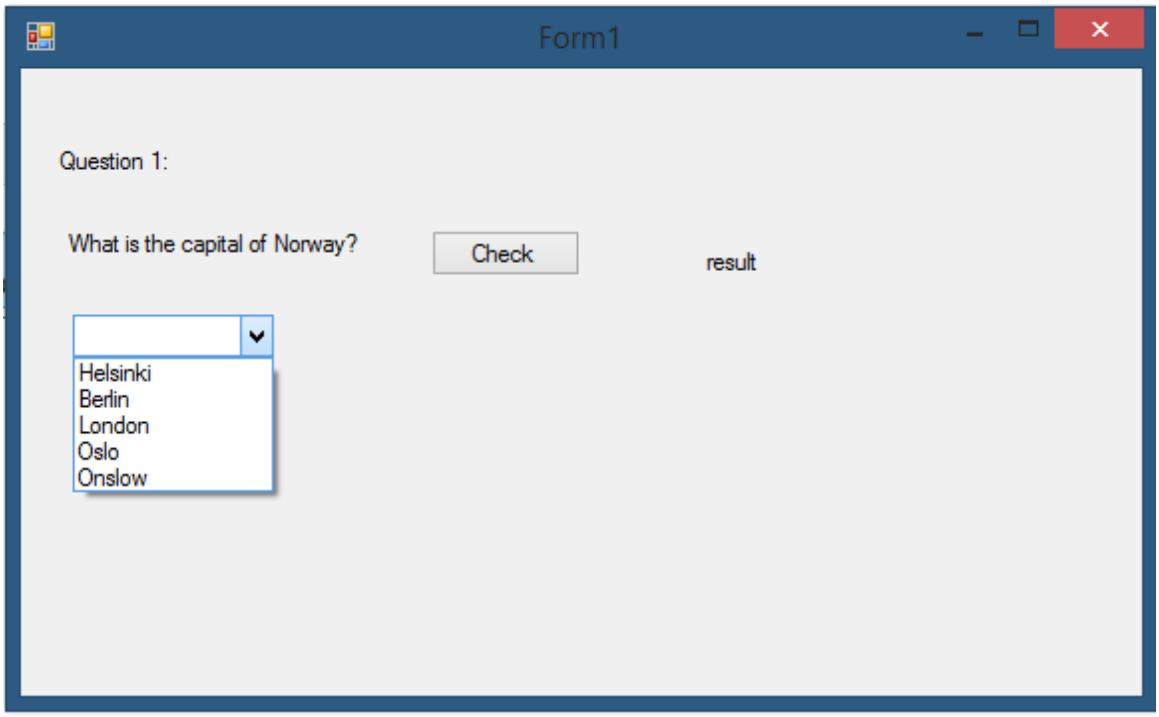
4

when user clicks the button, new dice value is generated. Amount of different dice numbers are shown as diagrams below.


Form1
Throw dice
Amounts of values
1: xxx
2: xxxxx
3: xx
4: xxxxx
5: x
6: xxx

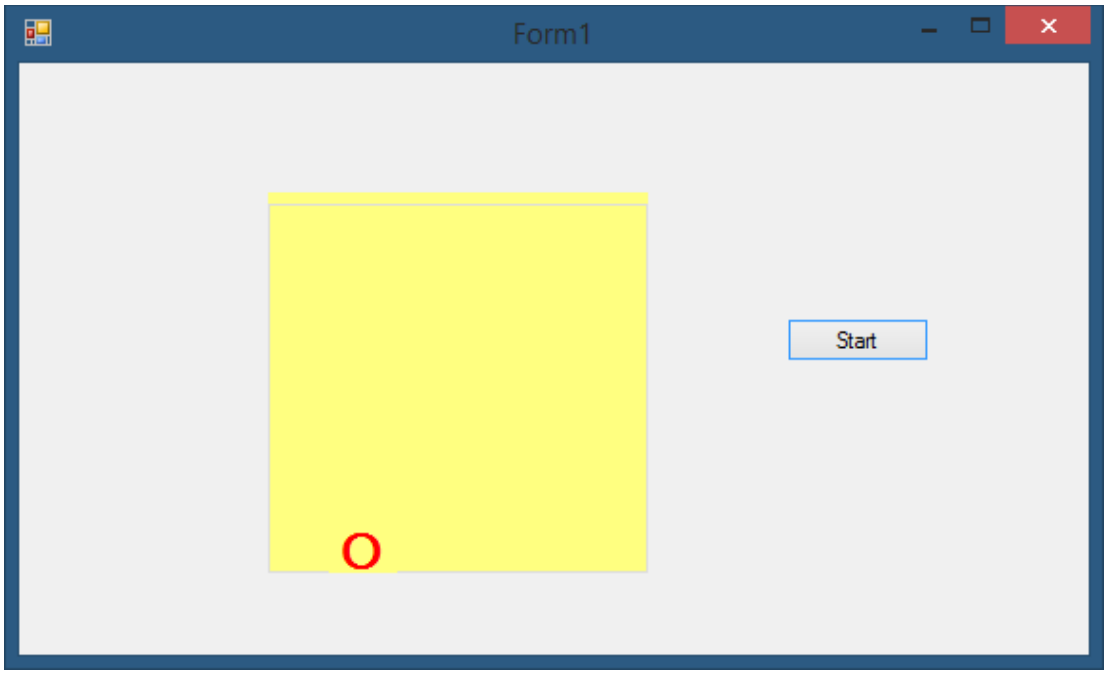
5

A question is generated to a label. User chooses from the list. With Check button answer is checked.



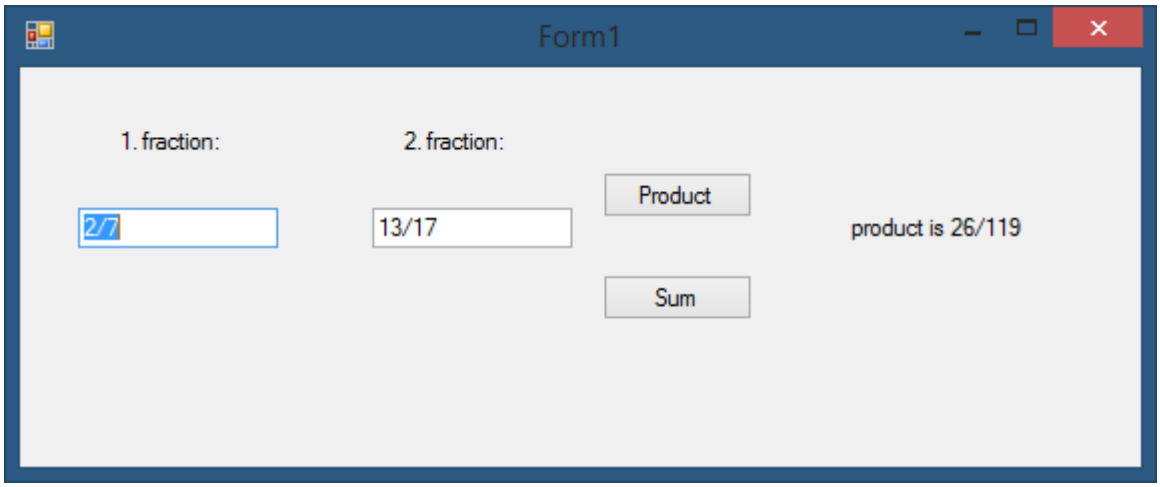
6

When button Start is clicked, a red circle starts to move along the sides of a square:



7

2 fractions are entered to textboxes. With buttons are sum and product calculated.



8

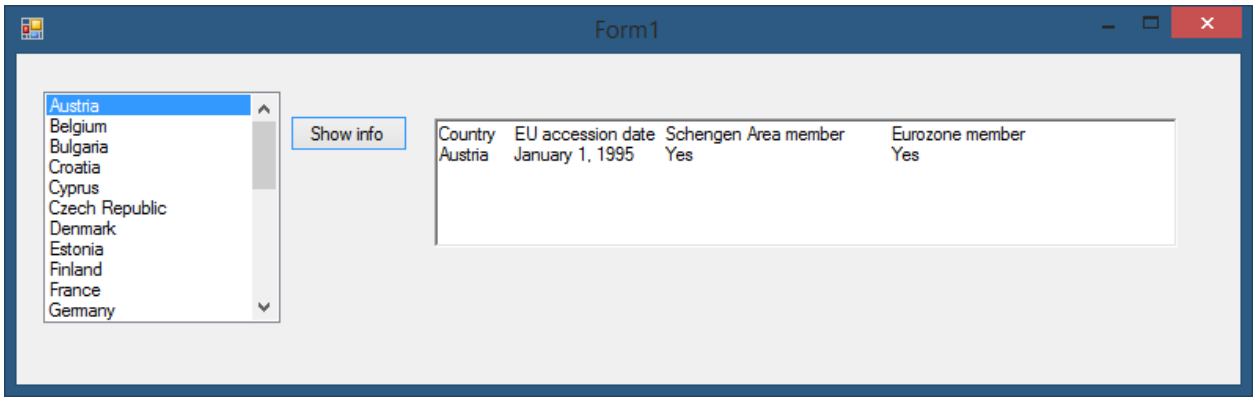
Program uses an array of EU countries:

Country	EU accession date	Schengen Area member	Eurozone member

```
  String[,] eu =
            {
{"Austria","January 1, 1995","Yes","Yes"},
{"Belgium","March 25, 1957","Yes","Yes"},
{"Bulgaria","January 1, 2007","No","No"},
{"Croatia","July 1, 2013","No","No"},
{"Cyprus","May 1, 2004","No","Yes"},
{"Czech Republic","May 1, 2004","Yes","No"},
{"Denmark","January 1, 1973","Yes","No"},
{"Estonia","May 1, 2004","Yes","Yes"},
{"Finland","January 1, 1995","Yes","Yes"},
{"France","March 25, 1957","Yes","Yes"},
{"Germany","March 25, 1957","Yes","Yes"},
{"Greece","January 1, 1981","Yes","Yes"},
{"Hungary","May 1, 2004","Yes","No"},
{"Ireland","January 1, 1973","No","Yes"},
{"Italy","March 25, 1957","Yes","Yes"},
{"Latvia","May 1, 2004","Yes","Yes"},
{"Lithuania","May 1, 2004","Yes","Yes"},
{"Luxembourg","March 25, 1957","Yes","Yes"},
{"Malta","May 1, 2004","Yes","Yes"},
{"Netherlands","March 25, 1957","Yes","Yes"},
```

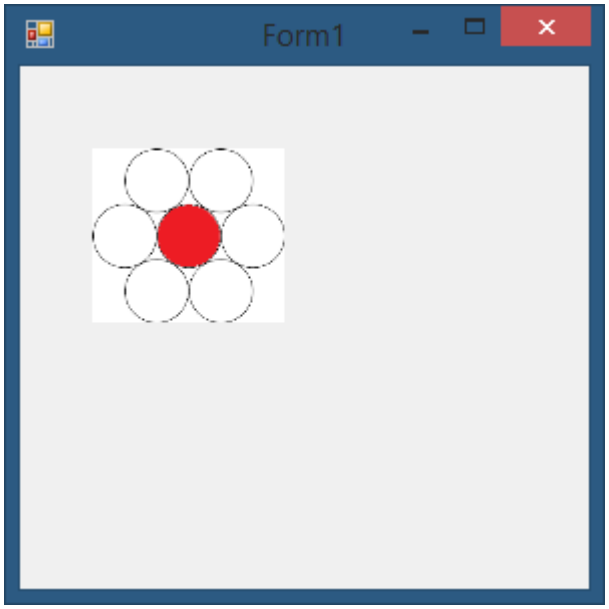
```
{"Poland","May 1, 2004","Yes","No"},
{"Portugal","January 1, 1986","Yes","Yes"},
{"Romania","January 1, 2007","No","No"},
{"Slovakia","May 1, 2004","Yes","Yes"},
{"Slovenia","May 1, 2004","Yes","Yes"},
{"Spain","January 1, 1986","Yes","Yes"},
{"Sweden","January 1, 1995","Yes","No"},
{"United Kingdom","January 1, 1973","No","No"}};
```

User chooses the country name from he list and button Show Info prints information about that country to a rich text control.



9

Create this animation:



Circles are colored red one by one. AT the end animation the middle circle is red.

19

By clicking a button 4 cards are shown on the form. Cards are taken from the shuffled deck.

11

Use at least one own function here:

Program calculates the amount of k combinations when k and n are given.

Example and formula:

$$\binom{52}{5} = \frac{n!}{k!(n-k)!} = \frac{52!}{5!(52-5)!} = \frac{52!}{5!47!} = 2,598,960.$$

! means factorial

That's all folks!

This is version 0.1 of the ebook "Basics in programming"

Product is to be completed step by step and all the feedback is taken into account!

This if free version!

Comments can be sent to darry.robinson@gmail.com

Youtube channel
https://www.youtube.com/@adamhigherstein8986

will introduce this ebook later…

Thank You!