

# Essential C#



# INSTRUCTOR INFORMATION

Loy Vanich  
084 007 5544  
Line ID: laployv  
laploy@gmail.com  
[www.laploy.com](http://www.laploy.com)



# TIME FRAME

- Course duration
- Start time
- Coffee break
- Lunch break
- Course end
- Parking

# Course materials

<https://github.com/laploy/C-101>

## Evaluation Form

<https://bit.ly/gfbiz-eval>

# Introduction to C#



# Creator

- Anders Hejlsberg
- Distinguished Engineer
- Developer Division
- Microsoft Corporation



---

# Big Ideas

- The first component oriented language in the C/C++ family
- Everything really is an object
- Next generation robust and durable software
- Preservation of investment

# Everything really is an object

- Traditional views
  - C++, Java: Primitive types are “magic” and do not interoperate with objects
  - Smalltalk, Lisp: Primitive types are objects, but at great performance cost
- C# unifies with no performance cost
  - Deep simplicity throughout system



# A component oriented language

- Component concepts are first class:
- Properties, methods, events
- Design-time and run-time attributes
- Integrated documentation using XML
- Enables one-stop programming
- No header files, IDL, etc.
- Can be embedded in web pages

---

# Robust and durable software

- Garbage collection
- No memory leaks and stray pointers
- Exceptions
- Error handling is not an afterthought
- Type-safety
- No uninitialized variables, unsafe casts
- Versioning
- Pervasive versioning considerations in aspects of language design

# Preservation of Investment

- C++ heritage
  - Namespaces, enums, unsigned types, pointers (in unsafe code), etc.
  - No unnecessary sacrifices
- Interoperability
  - What software is increasingly about
  - MS C# implementation talks to XML, SOAP, COM, DLLs, and any .NET language
- Millions of lines of C# code in .NET
  - Short learning curve
  - Increased productivity

# Hello world

```
using System;

class Hello
{
    static void Main() {
        Console.WriteLine("Hello world");
    }
}
```

# Program Structure

- Namespaces
  - Contain types and other namespaces
- Type declarations
  - Classes, structs, interfaces, enums, and delegates
- Members
  - Constants, fields, methods, properties, indexers, events, operators, constructors, destructors
- Organization
  - No header files, code written “in-line”
  - No declaration order dependence

# Type System

- Value types
  - Directly contain data
  - Cannot be null
- Reference types
  - Contain references to objects
  - May be null

```
int i = 123;  
string s = "Hello world"
```



# Type System

- Value types
  - Primitives
  - Enums
  - Structs
- Reference types
  - Classes
  - Interfaces
  - Arrays
  - Delegates

```
int i;  
enum State { Off, On }  
struct Point { int x, y; }
```

```
class Foo: Bar, IFoo {...}  
interface IFoo: IBar {...}  
string[] a = new string[10];  
delegate void Empty();
```

# Predefined Types

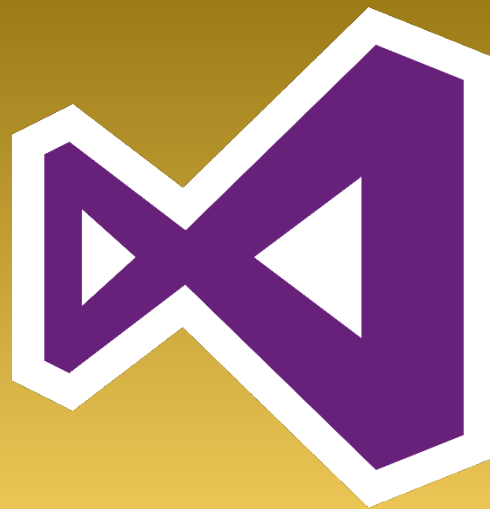
- C# predefined types
  - Reference                      object, string
  - Signed                        sbyte, short, int, long
  - Unsigned                    byte, ushort, uint, ulong
  - Character                    char
  - Floating-point              float, double, decimal
  - Logical                      bool
- Predefined types are simply aliases for system-provided types
  - For example, `int == System.Int32`



# Class

- Single inheritance
- Multiple interface implementation
- Class members
  - Constants, fields, methods, properties, indexers, events, operators, constructors, destructors
  - Static and instance members
  - Nested types
- Member access
  - public, protected, internal, private

# Visual Studio Basic



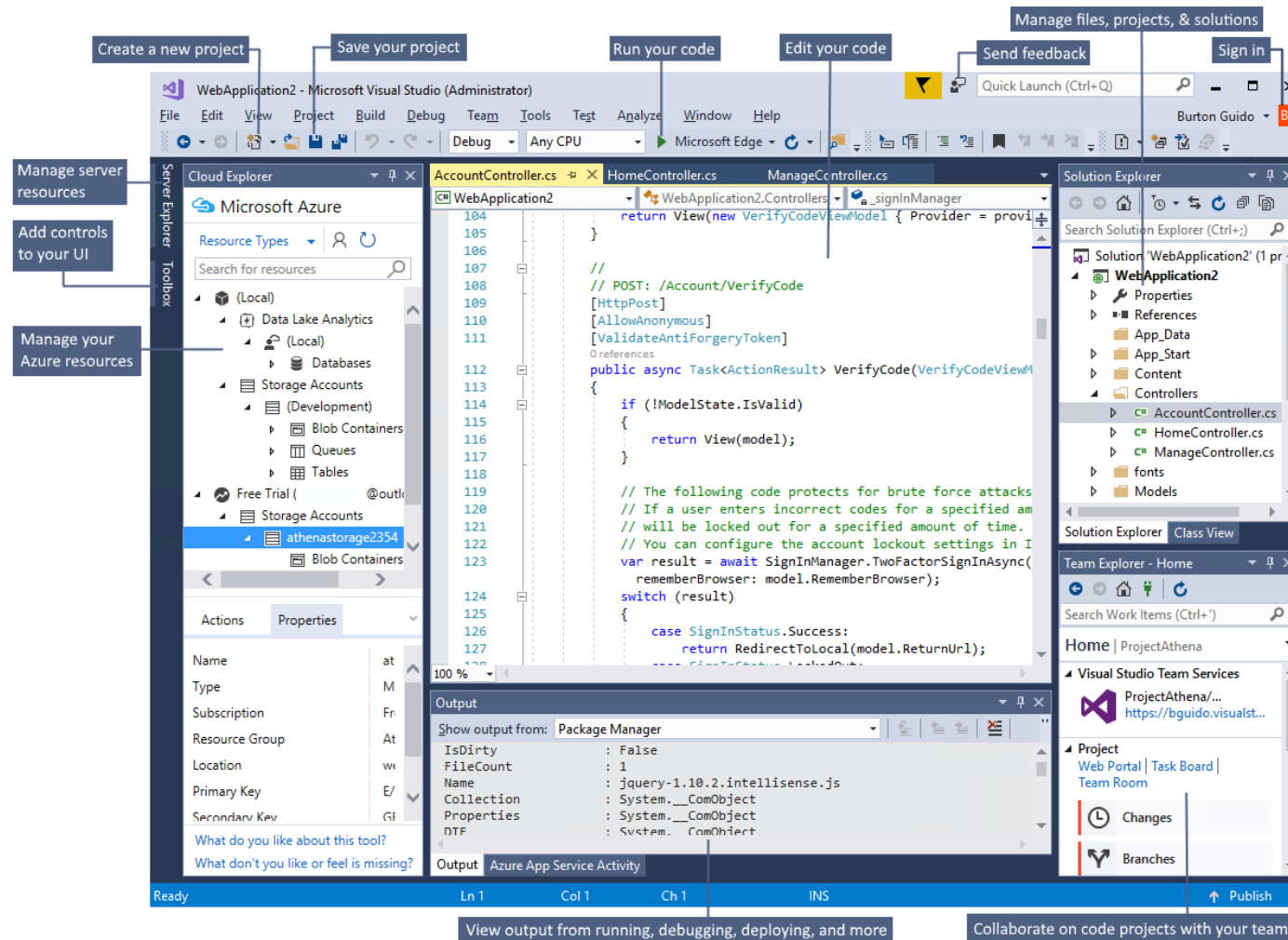
# Visual Studio Community 2017



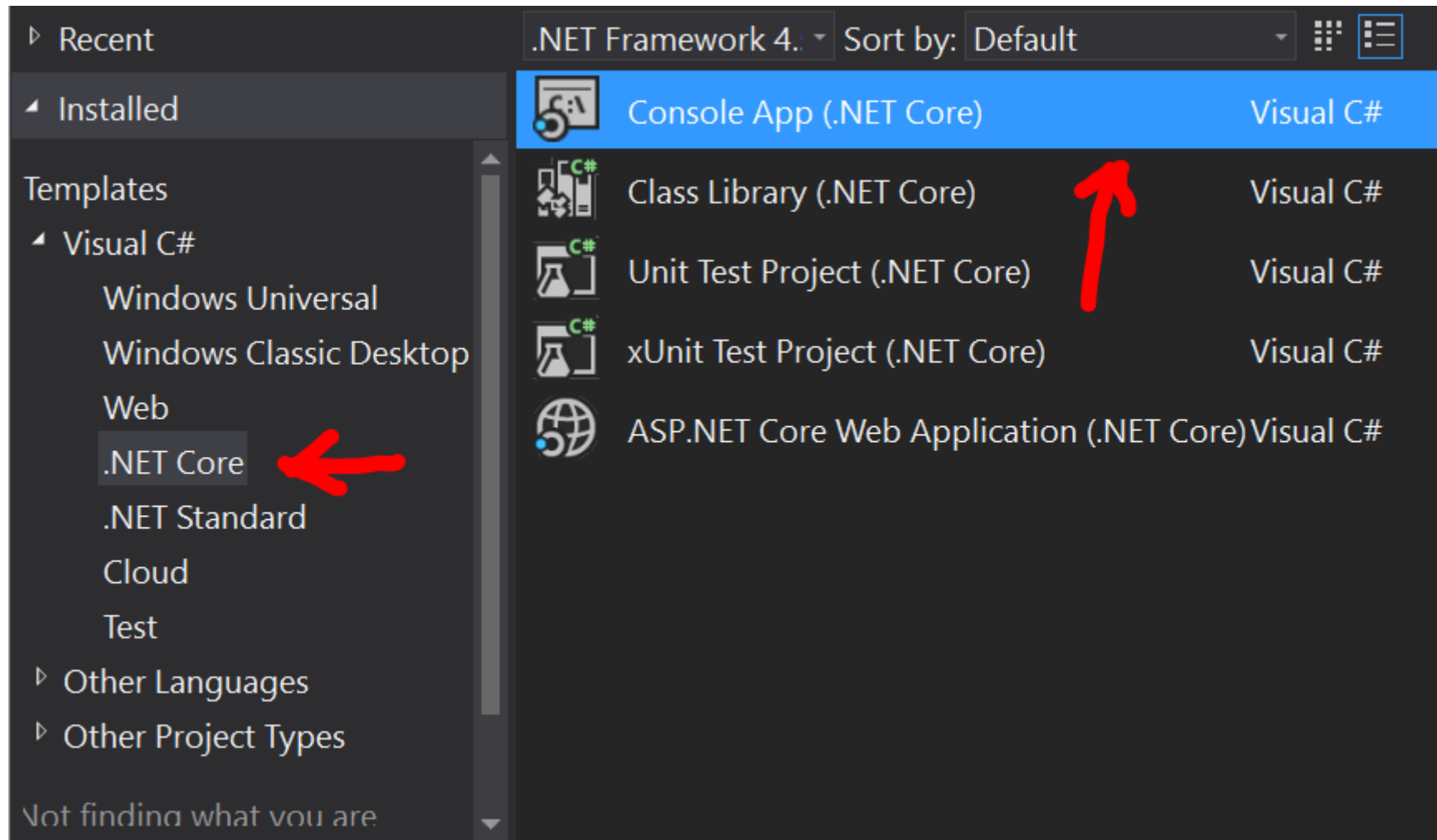
## Visual Studio IDE

Fully-featured integrated development environment (IDE)  
for Android, iOS, Windows, web, and cloud

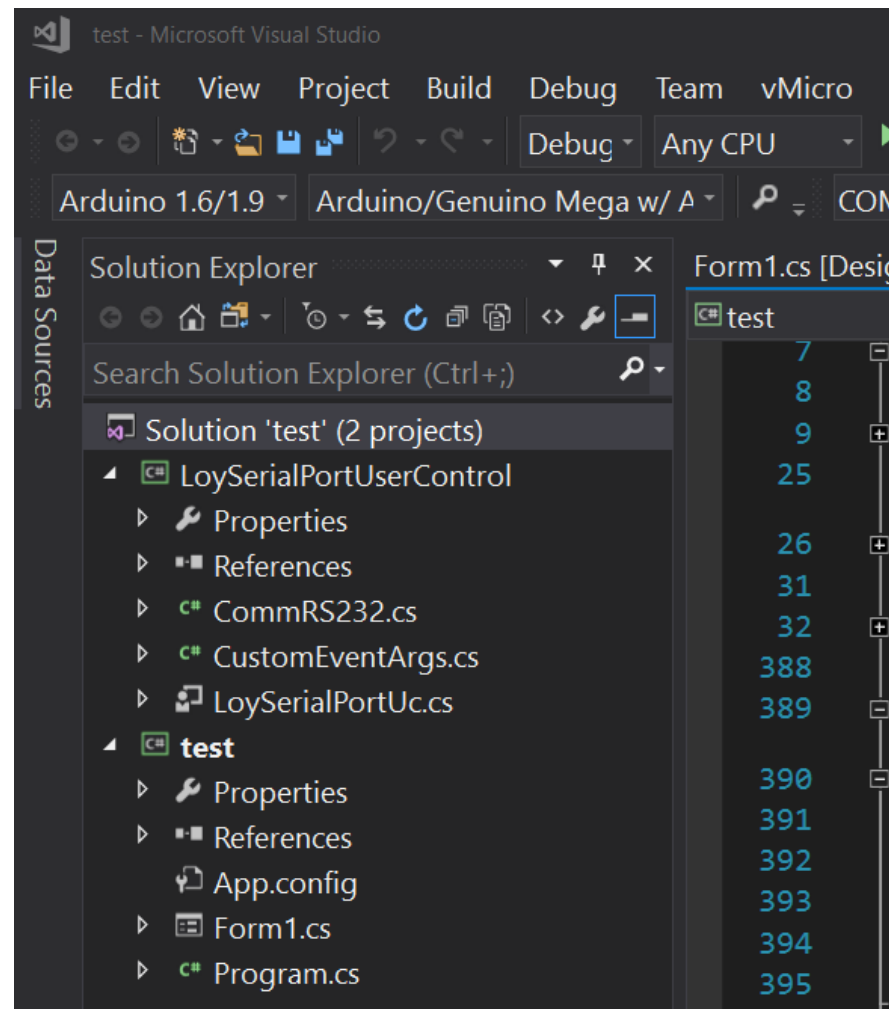
# Tour of the IDE



# Create New Project



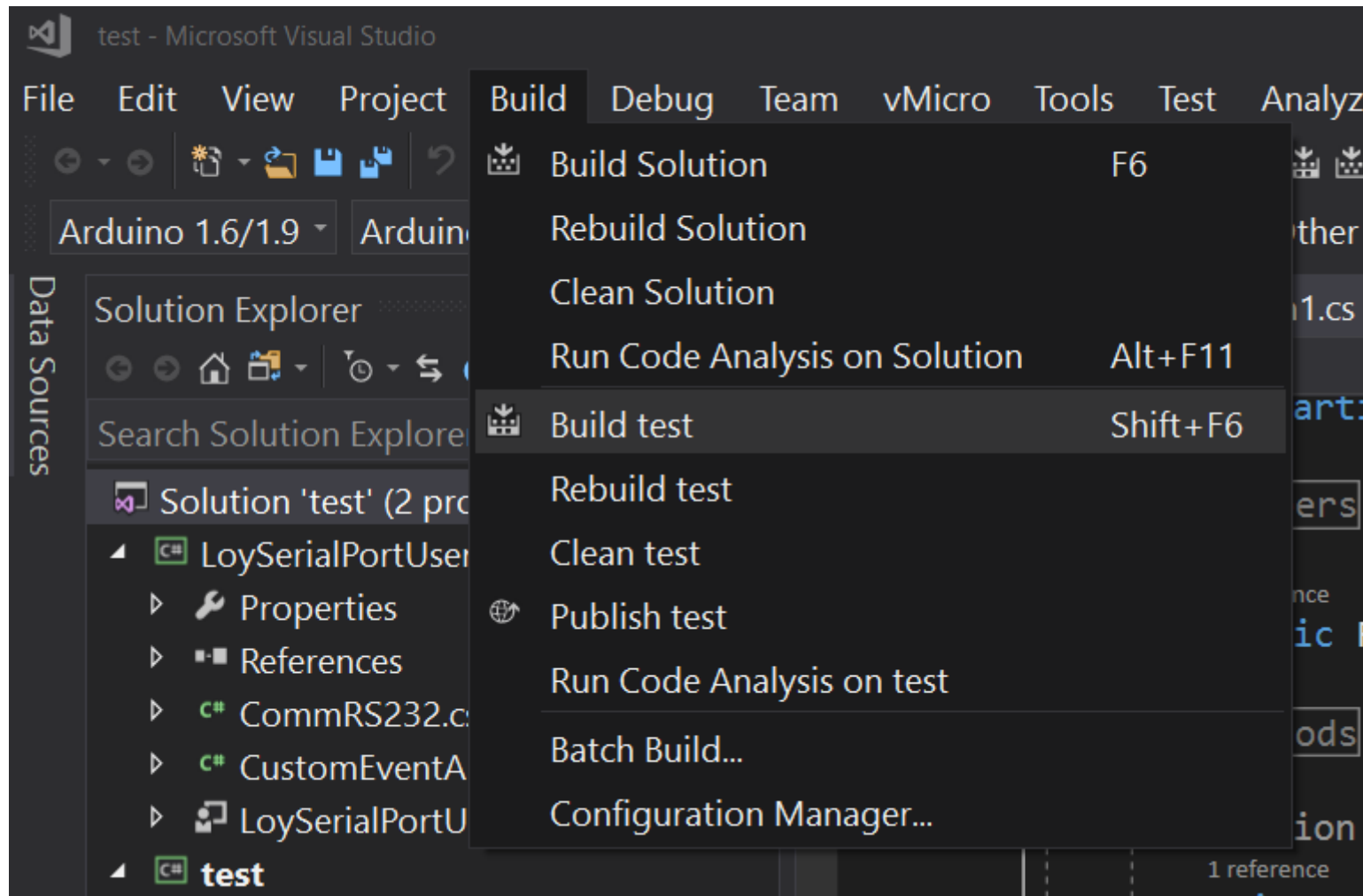
# Solution Explorer



# Code Entering

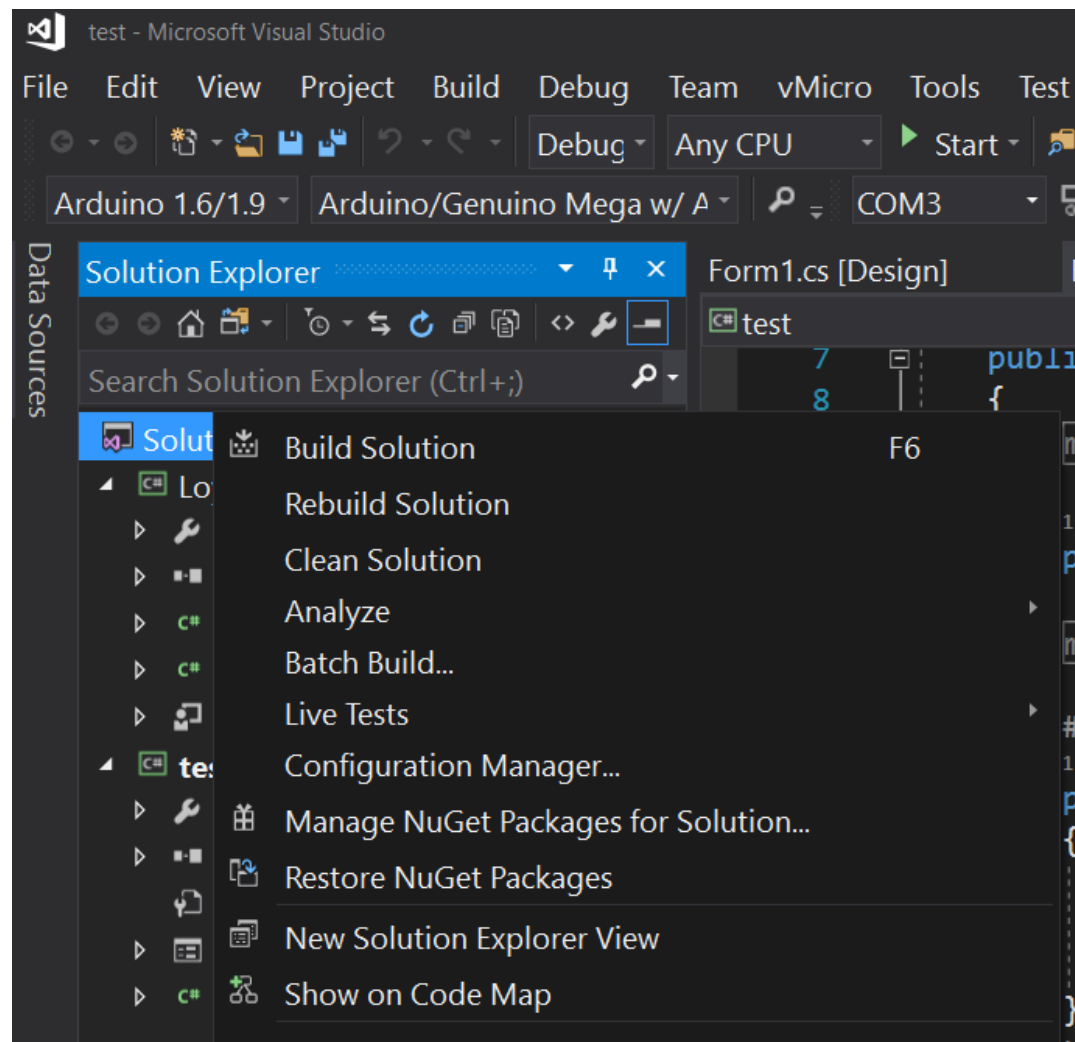
```
7 public partial class Form1 : Form
8 {
9     members
25
26     1 reference
    public Form1()...
31
32     methods
388
389     #region event
    1 reference
390     private void LoySerialPortUc1_OnDataReceived(string s)
391     {
392         ShowChar(s);
393         ShowHex(s);
394         loySerialPortUc1.Write("b");
395     }
    1 reference
396     private void ClearButtons()...
    1 reference
417     private void buttonClear_Click(object sender, EventArgs e)...
    1 reference
424     private void buttonSend_Click(object sender, EventArgs e)...
428     #endregion
429
```

# Building program

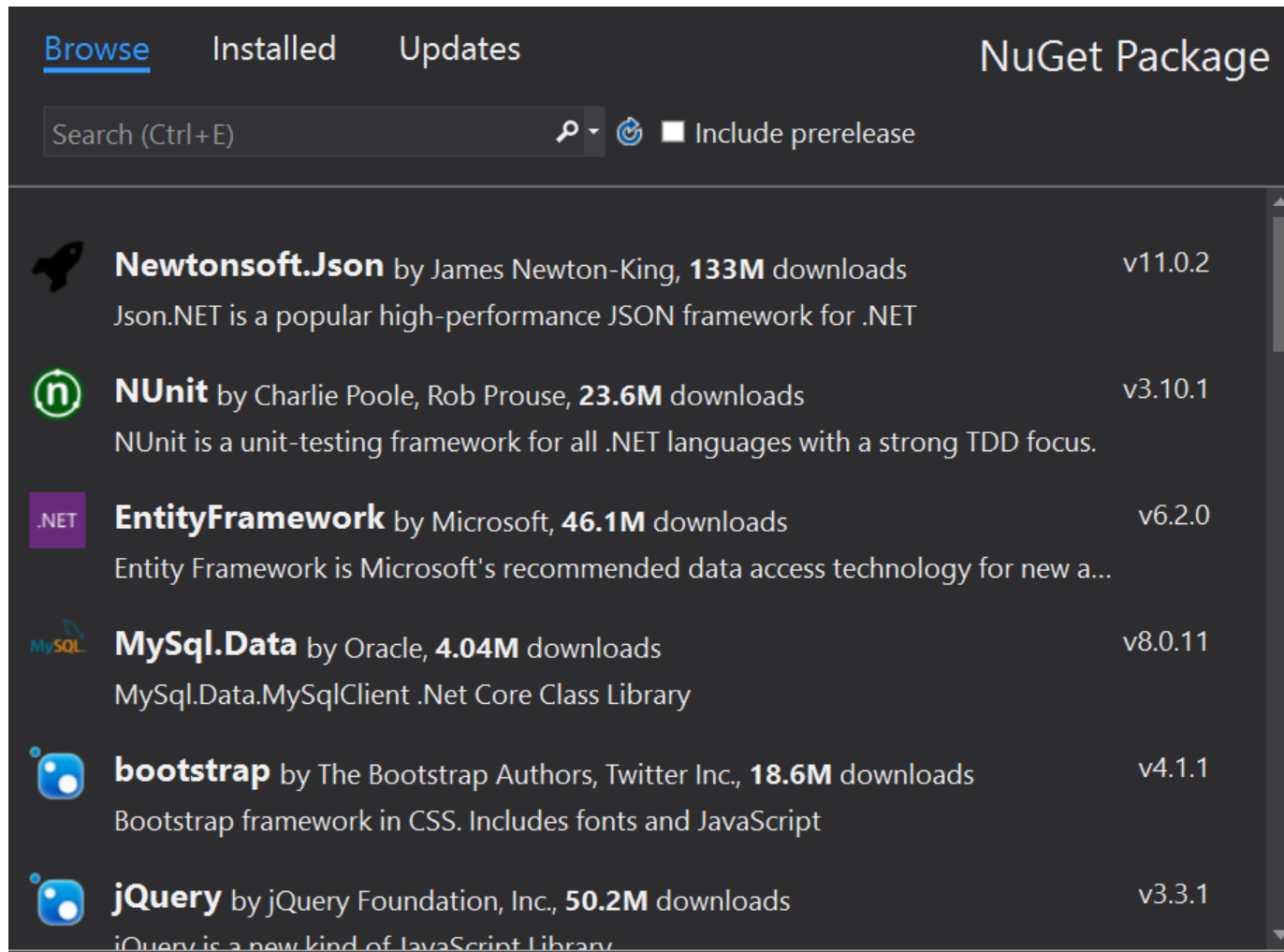




# Managing Solution / Project



# Managing Nuget package



# Excurses

1. Create a New project
2. Rename project / Solution
3. Build program and watch Output message
4. Run program and examine the result
5. Find the description of the `System.Console` class in the standard .NET API documentation (MSDN Library).
6. Find the description of the `System.Console.WriteLine()` method and its different possible parameters in the MSDN Library.

7. Compile and execute the sample program from this chapter using the command prompt (the console) and Visual Studio.
8. Modify the sample program to print a different greeting, for example "Good Day!".
9. Write a console application that prints your first and last name on the console.
10. Write a program that prints the following numbers on the console 1, 101, 1001, each on a new line.
11. Write a program that prints on the console the current date and time.
12. Write a program that prints the square root of 12345.

# Data Type



# C# Keywords

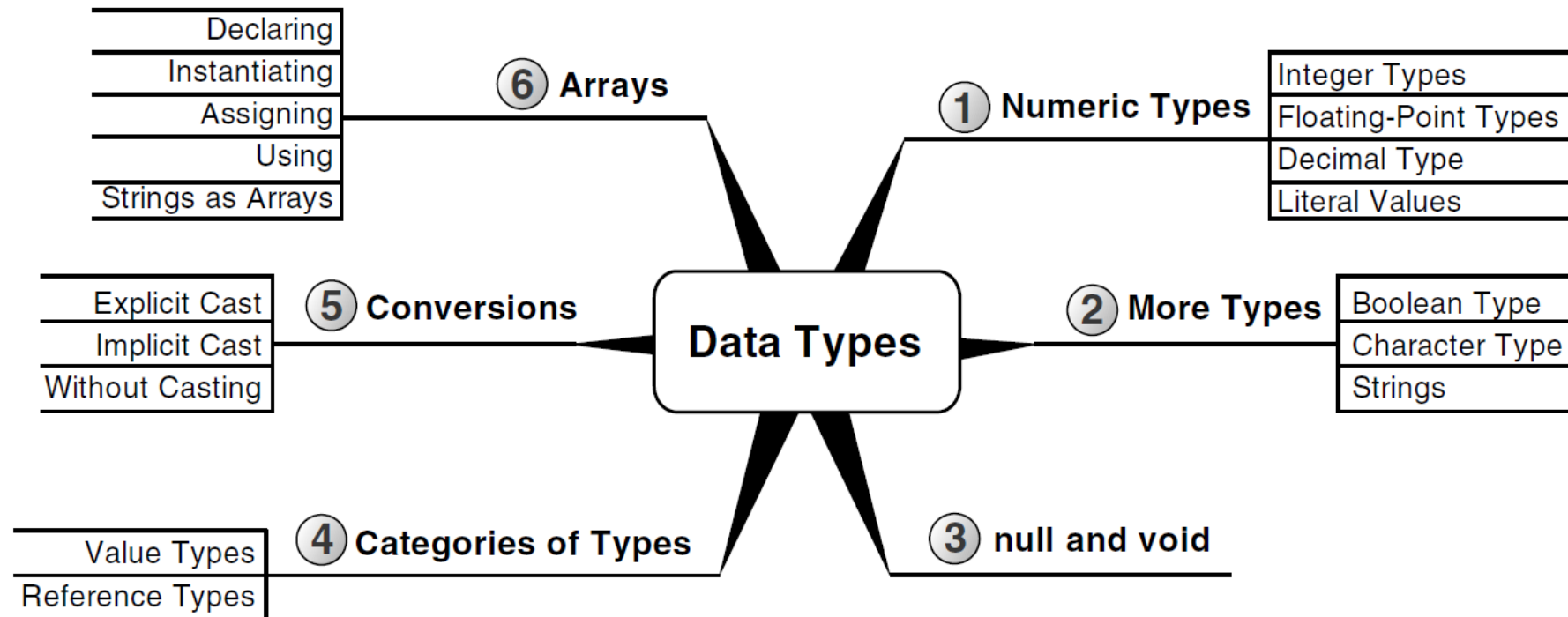
abstract	add*	as	ascending*
base	bool	break	by*
byte	case	catch	char
checked	class	const	continue
decimal	default	delegate	do
double	descending*	else	enum
event	explicit	extern	false
finally	fixed	from*	float
for	foreach	get*	group*
goto	if	implicit	in
int	into*	interface	internal
is	lock	long	join*
let*	namespace	new	null
object	operator	orderby*	out
override	params	partial*	private

# C# Keywords (cont')

<code>protected</code>	<code>public</code>	<code>readonly</code>	<code>ref</code>
<code>remove*</code>	<code>return</code>	<code>sbyte</code>	<code>sealed</code>
<code>select*</code>	<code>set*</code>	<code>short</code>	<code>sizeof</code>
<code>stackalloc</code>	<code>static</code>	<code>string</code>	<code>struct</code>
<code>switch</code>	<code>this</code>	<code>throw</code>	<code>true</code>
<code>try</code>	<code>typeof</code>	<code>uint</code>	<code>ulong</code>
<code>unchecked</code>	<code>unsafe</code>	<code>ushort</code>	<code>using</code>
<code>value*</code>	<code>virtual</code>	<code>void</code>	<code>volatile</code>
<code>where*</code>	<code>while</code>	<code>yield*</code>	

\* Contextual keyword

# Data Types





TYPE	SIZE	RANGE (INCLUSIVE)	BCL NAME	SIGNED
sbyte	8 bits	−128 to 127	System.SByte	Yes
byte	8 bits	0 to 255	System.Byte	No
short	16 bits	−32,768 to 32,767	System.Int16	Yes
ushort	16 bits	0 to 65,535	System.UInt16	No
int	32 bits	−2,147,483,648 to 2,147,483,647	System.Int32	Yes
uint	32 bits	0 to 4,294,967,295	System.UInt32	No
long	64 bits	−9,223,372,036,854,775,808 to 9,223,372,036,854,775,807	System.Int64	Yes
ulong	64 bits	0 to 18,446,744,073,709,551,615	System.UInt64	No

# Character Escape Sequences

Character	Escape Sequence name
\'	Single quote
\"	Double quote
\\	Backslash
\0	Null
\a	Alert
\b	Backspace
\f	Form feed
\n	New line
\r	Carriage return
\t	Horizontal tab
\v	Vertical quote

# Floating-point

TYPE	SIZE	RANGE (INCLUSIVE)	BCL NAME	SIGNIFICANT DIGITS
float	32 bits	$\pm 1.5 \times 10^{45}$ to $\pm 3.4 \times 10^{38}$	System.Single	7
double	64 bits	$\pm 5.0 \times 10^{324}$ to $\pm 1.7 \times 10^{308}$	System.Double	15–16

# Decimal

TYPE	SIZE	RANGE (INCLUSIVE)	BCL NAME	SIGNIFICANT DIGITS
decimal	128 bits	$1.0 \times 10^{-28}$ to approximately $7.9 \times 10^{28}$	System.Decimal	28–29

# Boolean Type

```
9      int a = 1; int b = 2;
10     // Which one is greater?
11     bool greaterAB = (a > b);
12     // Is 'a' equal to 1?
13     bool equalA1 = (a == 1);
14     // Print the results on the console
15     if (greaterAB)
16     {
17         Console.WriteLine("A > B");
18     }
19     else
20     {
21         Console.WriteLine("A <= B");
22     }
23     Console.WriteLine("greaterAB = " + greaterAB);
24     Console.WriteLine("equalA1 = " + equalA1);
25     // Console output:
26     // A <= B
27     // greaterAB = False
28     // equalA1 = True
```

# Strings

```
9      // Declare some variables
10     string firstName = "Loy";
11     string lastName = "Vanich";
12     string fullName = firstName + " " + lastName;
13     // Print the results on the console
14     Console.WriteLine("Hello, " + firstName + "!");
15     Console.WriteLine("Your full name is " + fullName + ".");
16     // Console output:
17     // Hello, Loy!
18     // Your full name is Loy Vanich.
19     Console.Read();
```

## Nullable Types

```
9      int i = 5;
10     int? ni = i;
11     Console.WriteLine(ni); // 5
12
13     // i = ni; // this will fail to compile
14     Console.WriteLine(ni.HasValue); // True
15     i = ni.Value;
16     Console.WriteLine(i); // 5
17
18     ni = null;
19     Console.WriteLine(ni.HasValue); // False
20     //i = ni.Value; // System.InvalidOperationException
21     i = ni.GetValueOrDefault();
22     Console.WriteLine(i); // 0
23     Console.Read();
```

# Variable Rules

- Variable names can contain the letters a-z, A-Z, the digits 0-9 as well as the character '\_'.
- Variable names cannot start with a digit.
- Variable names cannot coincide with a keyword of the C# language. For example, base, char, default, int, object, this, null and many others cannot be used as variable names.

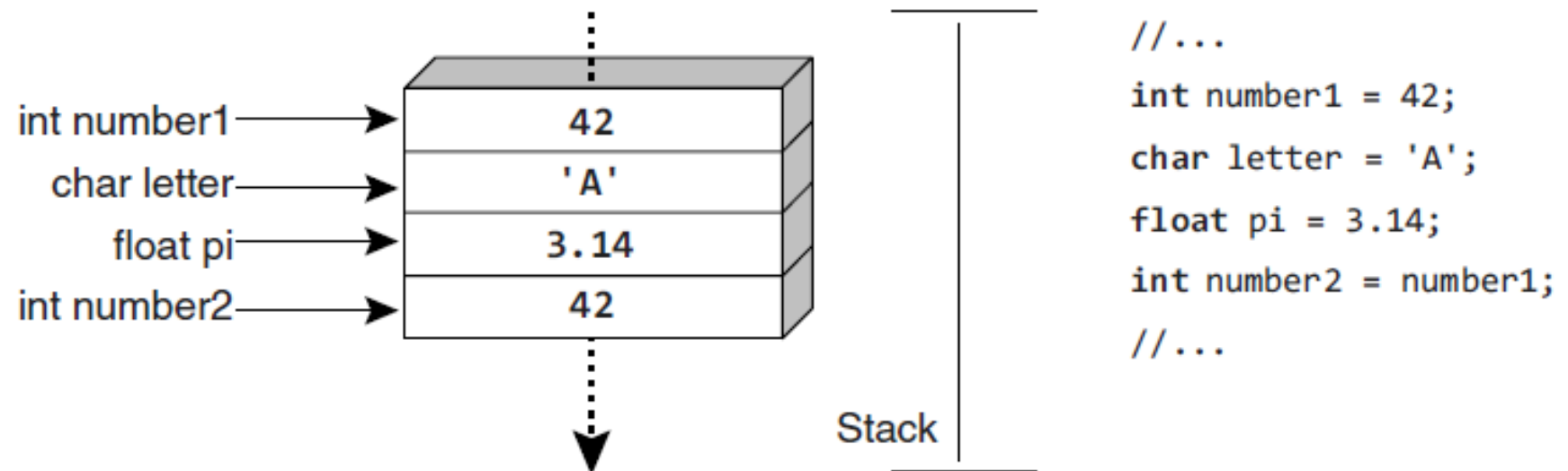


# Default Variable Values

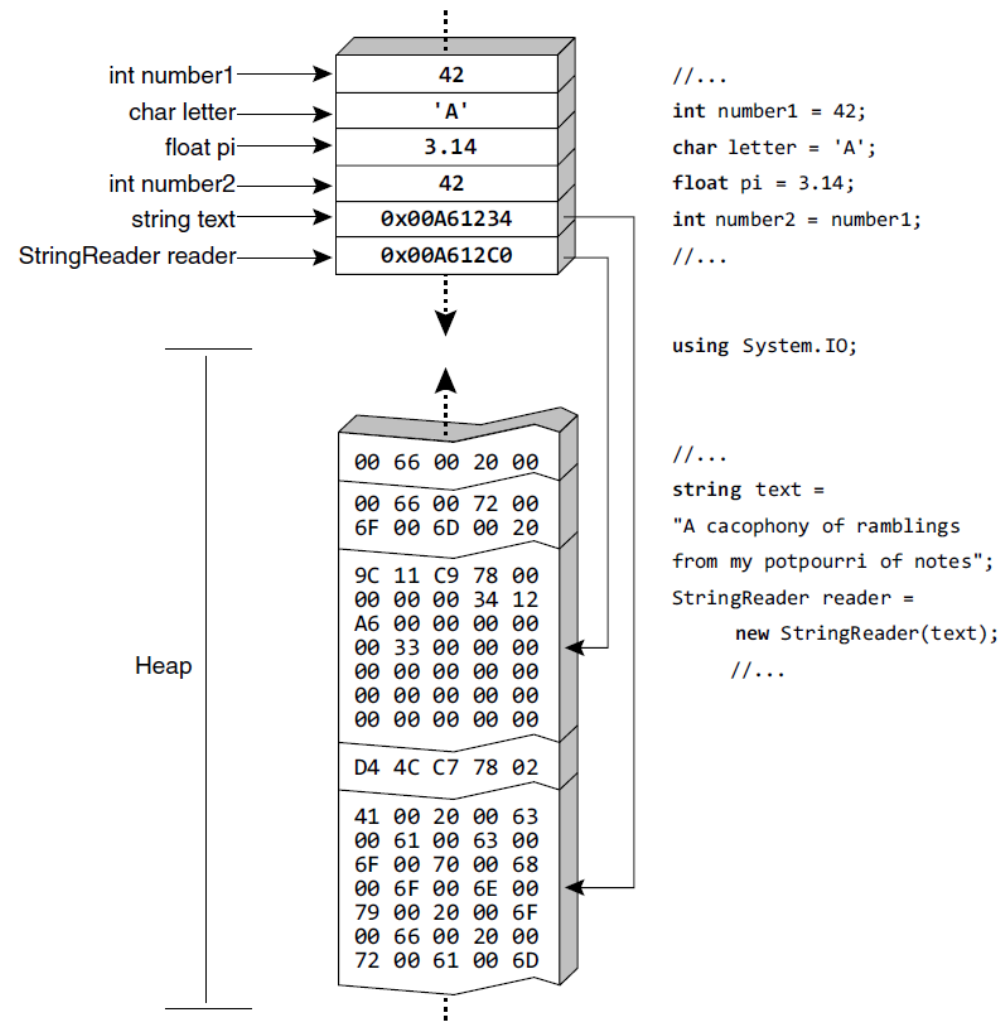
Data Type	Default Value
sbyte	0
byte	0
short	0
ushort	0
int	0
uint	0u
long	0L
ulong	0u

Data Type	Default Value
float	0.0f
double	0.0d
decimal	0.0m
bool	false
char	'\u0000'
string	null
object	null

# Value Type



# Reference Type



# Changing the Type of Data

- Widening and Narrowing
- Casting
- Type conversion
- Number < - > String conversion

# Array

- Array declaration
- Array value assignment
- Getting value from array
- Array iteration

# Literals

```
// An ordinary character
char character = 'a';
Console.WriteLine(character);
// Unicode character code in a hexadecimal format
character = '\u003A';
Console.WriteLine(character);
// Assigning the single quotation character (escaped as \')
character = '\'';
Console.WriteLine(character);
// Assigning the backslash character (escaped as \\)
character = '\\';
Console.WriteLine(character);
string quotation = "\"Hello, Jude\", he said.";
Console.WriteLine(quotation);
string path = "C:\\Windows\\Notepad.exe";
Console.WriteLine(path);
string verbatim = @"The \ is not escaped as \\. I am at a new line.";
Console.WriteLine(verbatim);
int myHex = 0x10;
Console.WriteLine(myHex);
```

# Exercise

1. Write a program that reads your age from the console and prints your age after 10 years.
2. User Console.WriteLine to Beep 5 times
3. Declare several variables by selecting for each one of them the most appropriate of the types sbyte, byte, short, ushort, int, uint, long and ulong in order to assign them the following values: 52,130; -115; 4825932; 97; -10000; 20000; 224; 970,700,000; 112; -44; -1,000,000; 1990; 123456789123456789.

- 
4. Initialize a variable of type `int` with a value of 256 in hexadecimal format (256 is 100 in a numeral system with base 16).
  5. Declare a variable of type `char` and assign it as a value the character, which has Unicode code, 72 (use the Windows calculator in order to find hexadecimal representation of 72).
  6. Declare a variable `isMale` of type `bool` and assign a value to it depending on your gender.



- 
7. Declare two variables of type string with values "Hello" and "World". Declare a variable of type object. Assign the value obtained of concatenation of the two string variables (add space if necessary) to this variable. Print the variable of type object.

- 
8. Declare two variables of type string and give them values "Hello" and "World". Assign the value obtained by the concatenation of the two variables of type string (do not miss the space in the middle) to a variable of type object. Declare a third variable of type string and initialize it with the value of the variable of type object (you should use type casting).

- 
9. Declare two variables of type string and assign them a value "The "use" of quotations causes difficulties." (without the outer quotes). In one of the variables use quoted string and in the other do not use it.
  10. Write a program to print a figure in the shape of a heart and others
  11. Declare two variables of type int. Assign to them values 5 and 10 respectively. Exchange (swap) their values and print them.

# Operators and Expressions

# Operator Categories

Category	Operators
arithmetic	<code>-</code> , <code>+</code> , <code>*</code> , <code>/</code> , <code>%</code> , <code>++</code> , <code>--</code>
logical	<code>&amp;&amp;</code> , <code>  </code> , <code>!</code> , <code>^</code>
binary	<code>&amp;</code> , <code> </code> , <code>^</code> , <code>~</code> , <code>&lt;&lt;</code> , <code>&gt;&gt;</code>
comparison	<code>==</code> , <code>!=</code> , <code>&gt;</code> , <code>&lt;</code> , <code>&gt;=</code> , <code>&lt;=</code>
assignment	<code>=</code> , <code>+=</code> , <code>-=</code> , <code>*=</code> , <code>/=</code> , <code>%=</code> , <code>&amp;=</code> , <code> =</code> , <code>^=</code> , <code>&lt;&lt;=</code> , <code>&gt;&gt;=</code>
string concatenation	<code>+</code>
type conversion	<code>(type)</code> , <code>as</code> , <code>is</code> , <code>typeof</code> , <code>sizeof</code>
other	<code>.</code> , <code>new</code> , <code>()</code> , <code>[]</code> , <code>?:</code> , <code>??</code>

# Types by Number of Arguments

Operator type	Number of arguments (operands)
unary	takes one operand
binary	takes two operands
ternary	takes three operands

# Operator Precedence

Priority	Operators
Highest priority     ...	(, )
	++, -- (as postfix), new, (type), typeof, sizeof
	++, -- (as prefix), +, - (unary), !, ~
	*, /, %
	+ (string concatenation)
	+, -
	<<, >>
	<, >, <=, >=, is, as
	==, !=
	&, ^,

# Operator Precedence

Lowest priority	&&
	?:, ??
	=, *=, /=, %=, +=, -=, <<=, >>=, &=, ^=,  =



# Arithmetical Operators – Example

```
// Arithmetical Operators - Example
int squarePerimeter = 17;
double squareSide = squarePerimeter / 4.0;
double squareArea = squareSide * squareSide;
Console.WriteLine(squareSide); // 4.25
Console.WriteLine(squareArea); // 18.0625
int a = 5;
int b = 4;
Console.WriteLine(a + b); // 9
Console.WriteLine(a + (b++)); // 9
Console.WriteLine(a + b); // 10
Console.WriteLine(a + (++b)); // 11
Console.WriteLine(a + b); // 11
Console.WriteLine(14 / a); // 2
Console.WriteLine(14 % a); // 4
int one = 1;
int zero = 0;
// Console.WriteLine(one / zero); // DivideByZeroException
double dMinusOne = -1.0;
double dZero = 0.0;
Console.WriteLine(dMinusOne / zero); // -Infinity
Console.WriteLine(one / dZero); // Infinity
```

# Logical Operators

x	y	!x	x && y	x    y	x ^ y
true	true	false	true	true	false
true	false	false	false	true	true
false	true	true	false	true	true
false	false	true	false	false	false

```
bool a = true;
bool b = false;
Console.WriteLine(a && b); // False
Console.WriteLine(a || b); // True
Console.WriteLine(!b); // True
Console.WriteLine(b || true); // True
Console.WriteLine((5 > 7) ^ (a == b)); // False
```

# Bitwise Operators

x	y	~x	x & y	x   y	x ^ y
1	1	0	1	1	0
1	0	0	0	1	1
0	1	1	0	1	1
0	0	1	0	0	0

```
byte a = 3; // 0000 0011 = 3
byte b = 5; // 0000 0101 = 5
Console.WriteLine(a | b); // 0000 0111 = 7
Console.WriteLine(a & b); // 0000 0001 = 1
Console.WriteLine(a ^ b); // 0000 0110 = 6
Console.WriteLine(~a & b); // 0000 0100 = 4
Console.WriteLine(a << 1); // 0000 0110 = 6
Console.WriteLine(a << 2); // 0000 1100 = 12
Console.WriteLine(a >> 1); // 0000 0001 = 1
```

# Comparison Operators

```
int x = 10, y = 5;  
Console.WriteLine("x > y : " + (x > y)); // True  
Console.WriteLine("x < y : " + (x < y)); // False  
Console.WriteLine("x >= y : " + (x >= y)); // True  
Console.WriteLine("x <= y : " + (x <= y)); // False  
Console.WriteLine("x == y : " + (x == y)); // False  
Console.WriteLine("x != y : " + (x != y)); // True
```

# Other Operator

```
int a = 6; int b = 3;  
Console.WriteLine(a + b / 2); // 7  
Console.WriteLine((a + b) / 2); // 4  
string s = "Beer";  
Console.WriteLine(s is string); // True  
string notNullString = s;  
string nullString = null;  
Console.WriteLine(nullString ?? "Unspecified"); // Unspecified  
Console.WriteLine(notNullString ?? "Specified"); // Beer
```

# Expression

```
int r = (150 - 20) / 2 + 5;
// Expression for calculating the surface of the circle
double surface = Math.PI * r * r;
// Expression for calculating the perimeter of the circle
double perimeter = 2 * Math.PI * r;
Console.WriteLine(r);
Console.WriteLine(surface);
Console.WriteLine(perimeter);
// use bracket to make the code clear
double incorrect = (double)((1 + 2) / 4);
Console.WriteLine(incorrect); // 0

double correct = ((double)(1 + 2)) / 4;
Console.WriteLine(correct); // 0.75

Console.WriteLine("2 + 3 = " + 2 + 3); // 2 + 3 = 23
Console.WriteLine("2 + 3 = " + (2 + 3)); // 2 + 3 = 5
```

# Exercises

1. Write an expression that checks whether an integer is odd or even.
2. Write a Boolean expression that checks whether a given integer is divisible by both 5 and 7, without a remainder.
3. Write an expression that looks for a given integer if its third digit (right to left) is 7.
4. Write an expression that checks whether the third bit in a given integer is 1 or 0.
5. Write an expression that calculates the area of a trapezoid by given sides  $a$ ,  $b$  and height  $h$ .
6. Write a program that prints on the console the perimeter and the area of a rectangle by given side and height entered by the user.
7. The gravitational field of the Moon is approximately 17% of that on the Earth. Write a program that calculates the weight of a man on the moon by a given weight on the Earth.
8. Write an expression that checks for a given point  $\{x, y\}$  if it is within the circle  $K(\{0, 0\}, R=5)$ .  
Explanation: the point  $\{0, 0\}$  is the center of the circle and 5 is the radius.

9. Write an expression that checks for given point  $\{x, y\}$  if it is within the circle  $K(\{0, 0\}, R=5)$  and out of the rectangle  $[-1, 1], [5, 5]$ . Clarification: for the rectangle the lower left and the upper right corners are given.

10. Write a program that takes as input a four-digit number in format  $abcd$  (e.g. 2011) and performs the following actions:

- Calculates the sum of the digits (in our example  $2+0+1+1 = 4$ ).
- Prints on the console the number in reversed order:  $dcba$  (in our example 1102).
- Puts the last digit in the first position:  $dabc$  (in our example 1201).
- Exchanges the second and the third digits:  $acbd$  (in our example 2101).

11. We are given a number  $n$  and a position  $p$ . Write a sequence of operations that prints the value of the bit on the position  $p$  in the number (0 or 1). Example:  $n=35, p=5 \rightarrow 1$ . Another example:  $n=35, p=6 \rightarrow 0$ .

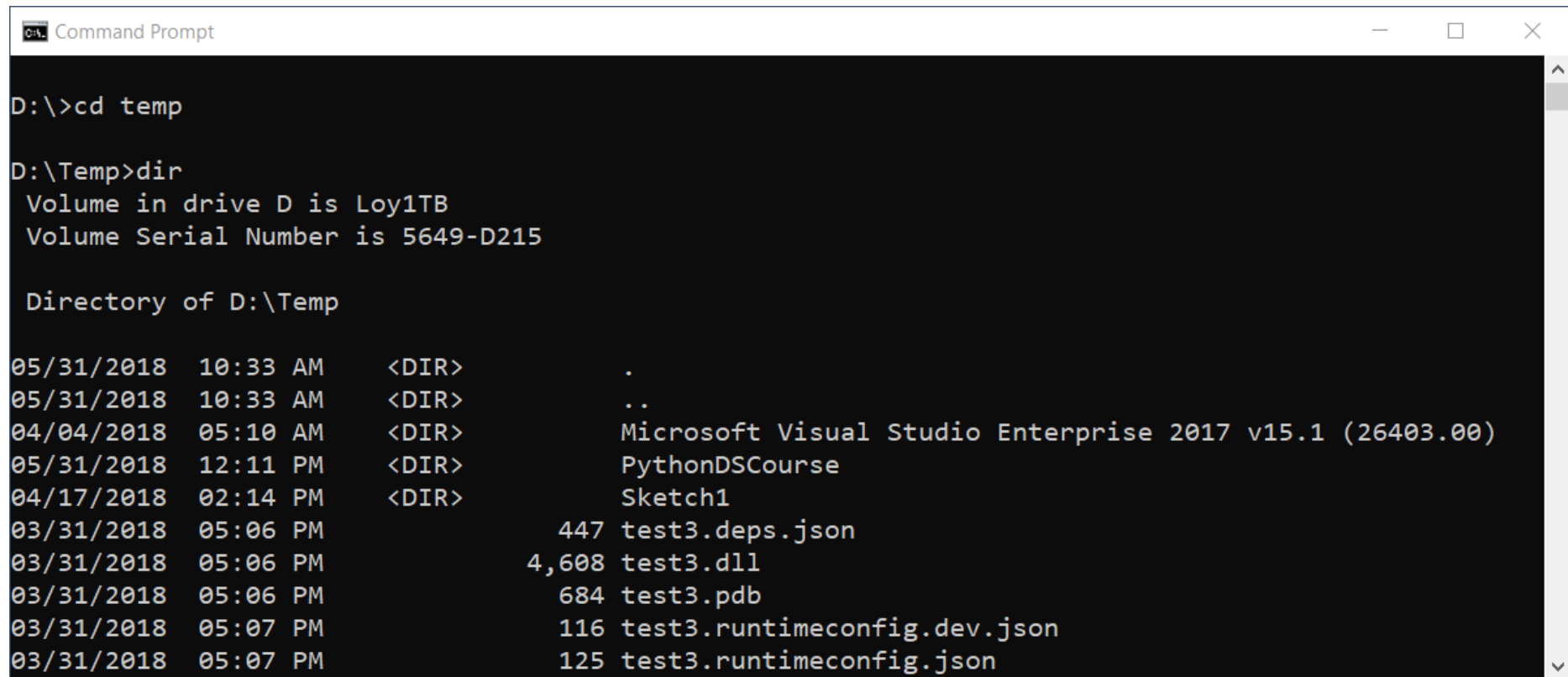
12. Write a Boolean expression that checks if the bit on position  $p$  in the integer  $v$  has the value 1. Example  $v=5, p=1 \rightarrow \text{false}$ .



- 
13. We are given the number  $n$ , the value  $v$  ( $v = 0$  or  $1$ ) and the position  $p$ . write a sequence of operations that changes the value of  $n$ , so the bit on the position  $p$  has the value of  $v$ . Example:  $n=35, p=5, v=0 \rightarrow n=3$ . Another example:  $n=35, p=2, v=1 \rightarrow n=39$ .
14. Write a program that checks if a given number  $n$  ( $1 < n < 100$ ) is a prime number (i.e. it is divisible without remainder only to itself and 1).
15. \* Write a program that exchanges the values of the bits on positions 3, 4 and 5 with bits on positions 24, 25 and 26 of a given 32-bit unsigned integer.
16. \* Write a program that exchanges bits  $\{p, p+1, \dots, p+k-1\}$  with bits  $\{q, q+1, \dots, q+k-1\}$  of a given 32-bit unsigned integer.

# Console Input and Output

# Invoking console



```
Command Prompt

D:\>cd temp

D:\Temp>dir
Volume in drive D is Loy1TB
Volume Serial Number is 5649-D215

Directory of D:\Temp

05/31/2018  10:33 AM    <DIR>          .
05/31/2018  10:33 AM    <DIR>          ..
04/04/2018  05:10 AM    <DIR>          Microsoft Visual Studio Enterprise 2017 v15.1 (26403.00)
05/31/2018  12:11 PM    <DIR>          PythonDSCourse
04/17/2018  02:14 PM    <DIR>          Sketch1
03/31/2018  05:06 PM             447 test3.deps.json
03/31/2018  05:06 PM          4,608 test3.dll
03/31/2018  05:06 PM             684 test3.pdb
03/31/2018  05:07 PM             116 test3.runtimeconfig.dev.json
03/31/2018  05:07 PM             125 test3.runtimeconfig.json
```

# Basic commands

Command	Description
<code>dir</code>	Displays the content of the current directory.
<code>cd &lt;directory name&gt;</code>	Changes the current directory.
<code>mkdir &lt;directory name&gt;</code>	Creates a new directory in the current one.
<code>rmdir &lt;directory name&gt;</code>	Deletes an existing directory.
<code>type &lt;file name&gt;</code>	Prints file content.
<code>copy &lt;src file&gt; &lt;destination file&gt;</code>	Copies one file into another.

# Console.WriteLine

```
Console.Out.WriteLine("Hello World");  
// Print String  
Console.WriteLine("Hello World");  
// Print int  
Console.WriteLine(5);  
// Print double  
Console.WriteLine(3.14159265358979);  
// multiple line / new line  
Console.WriteLine("I love");  
Console.Write("this ");  
Console.Write("Book!");  
// String concat  
string age = "twenty six";  
string text = "He is " + age + " years old.";  
Console.WriteLine(text);  
Console.WriteLine("He is " + age + " years old.");  
string s = "Four: " + 2 + 2;  
Console.WriteLine(s);  
// Four: 22  
string s1 = "Four: " + (2 + 2);  
Console.WriteLine(s1);  
// Four: 4
```

# String formatting

```
string str = "Hello World!";  
// Print (the normal way)  
Console.Write(str);  
// Print (through formatting string)  
Console.Write("{0}", str);  
string name = "John";  
int age = 18;  
string town = "Seattle";  
Console.Write("{0} is {1} years old from {2}!\n", name, age, town);  
Console.Write("{1} is {0} years old from {3}!", 18, "John", 0, "Seattle");  
// Alignment Component  
Console.WriteLine("{0,6}", 123);  
Console.WriteLine("{0,6}", 1234);  
Console.WriteLine("{0,6}", 12);  
Console.Write("{0,-6}", 123);  
Console.WriteLine("--end");
```

# Numbers formatting

```
// StandardNumericFormats
Console.WriteLine("{0:C2}", 123.456);
//Output: 123,46 лв.
Console.WriteLine("{0:D6}", -1234);
//Output: -001234
Console.WriteLine("{0:E2}", 123);
//Output: 1,23E+002
Console.WriteLine("{0:F2}", -123.456);
//Output: -123,46
Console.WriteLine("{0:N2}", 1234567.8);
//Output: 1 234 567,80
Console.WriteLine("{0:P}", 0.456);
//Output: 45,60 %
Console.WriteLine("{0:X}", 254);
//Output: FE
```

# Custom numeric format

```
// CustomNumericFormats
Console.WriteLine("{0:0.00}", 1);
//Output: 1.00
Console.WriteLine("{0:0.##}", 0.234);
//Output: .23
Console.WriteLine("{0:#####}", 12345.67);
//Output: 12346
Console.WriteLine("{0:(0#) ### ## ##}", 29342525);
//Output: (02) 934 25 25
Console.WriteLine("{0:%##}", 0.234);
//Output: %23
DateTime d = new DateTime(2012, 02, 27, 17, 30, 22);
Console.WriteLine("{0:dd/MM/yyyy HH:mm:ss}", d);
Console.WriteLine("{0:d.MM.yy}", d);
Console.WriteLine("{0:G}", DayOfWeek.Wednesday); // Wednesday
Console.WriteLine("{0:D}", DayOfWeek.Wednesday); // 3
Console.WriteLine("{0:X}", DayOfWeek.Wednesday); // 00000003
```



# ReadLine()

```
Console.Write("Please enter your first name: ");  
string firstName = Console.ReadLine();  
Console.Write("Please enter your last name: ");  
string lastName = Console.ReadLine();  
Console.WriteLine("Hello, {0} {1}!", firstName, lastName);  
// Output: Please enter your first name: John  
// Please enter your last name: Smith  
// Hello, John Smith!
```

# Read()

```
// using read()
int codeRead = 0;
do
{
    codeRead = Console.Read();
    if (codeRead != 0)
    {
        Console.Write((char)codeRead);
    }
}
while (codeRead != 10);
```

## Reading Number

```
// reading number
Console.Write("a = ");
int a = int.Parse(Console.ReadLine());
Console.Write("b = ");
int b = int.Parse(Console.ReadLine());
Console.WriteLine("{0} + {1} = {2}", a, b, a + b);
Console.WriteLine("{0} * {1} = {2}", a, b, a * b);
Console.Write("f = ");
double f = double.Parse(Console.ReadLine());
Console.WriteLine("{0} * {1} / {2} = {3}", a, b, f, a * b / f);
//a = 5
//b = 6
//5 + 6 = 11
//5 * 6 = 30
//f = 7.5
//5 * 6 / 7.5 = 4
```

# Pars input

```
// pars string to double
Console.Write("Enter a floating-point number: ");
string line = Console.ReadLine();
double number = double.Parse(line);
Console.WriteLine("You entered: {0}", number);
// Parsing Numbers Conditionally
string str = Console.ReadLine();
int intValue;
bool parseSuccess = Int32.TryParse(str, out intValue);
Console.WriteLine(parseSuccess ?
"The square of the number is " + intValue * intValue + "."
: "Invalid number!");
```

## ReadKey()

```
// Reading by Console.ReadKey()
ConsoleKeyInfo key = Console.ReadKey();
Console.WriteLine();
Console.WriteLine("Character entered: " + key.KeyChar);
Console.WriteLine("Special keys: " + key.Modifiers);
//A
//Character entered: A
//Special keys: Shift
```

# Example 1

```
// print out letter from book publisher to reader
Console.Write("Enter person name: ");
string person = Console.ReadLine();
Console.Write("Enter book name: ");
string book = Console.ReadLine();
string from = "Authors Team";
Console.WriteLine(" Dear {0},", person);
Console.Write("We are pleased to inform " +
"you that \"{1}\" is the best Bulgarian book. {2}" +
"The authors of the book wish you good luck {0}!{2}",
person, book, Environment.NewLine);
Console.WriteLine(" Yours,");
Console.WriteLine(" {0}", from);
//Enter person name: Readers
//Enter book name: Introduction to programming with C#
//Dear Readers,
//We are pleased to inform you that "Introduction to programming
//with C#" is the best Bulgarian book.
//The authors of the book wish you good luck Readers!
//Yours,
//Authors Team
```


## Example 2

```
Console.WriteLine("This program calculates " +  
    "the area of a rectangle or a triangle");  
Console.WriteLine("Enter a and b (for rectangle) " +  
    "or a and h (for triangle): ");  
int a = int.Parse(Console.ReadLine());  
int b = int.Parse(Console.ReadLine());  
Console.WriteLine("Enter 1 for a rectangle or " +  
    "2 for a triangle: ");  
int choice = int.Parse(Console.ReadLine());  
double area = (double)(a * b) / choice;  
Console.WriteLine("The area of your figure is " + area);  
  
//This program calculates the area of a rectangle or a triangle  
//Enter a and b(for rectangle) or a and h(for triangle):  
//5  
//4  
//Enter 1 for a rectangle or 2 for a triangle:  
//2  
//The area of your figure is 10
```

# Exercises

1. Write a program that reads from the console three numbers of type `int` and prints their sum.
2. Write a program that reads from the console the radius "r" of a circle and prints its perimeter and area.
3. A given company has name, address, phone number, fax number, web site and manager. The manager has name, surname and phone number. Write a program that reads information about the company and its manager and then prints it on the console.
4. Write a program that prints three numbers in three virtual columns on the console. Each column should have a width of 10 characters and the numbers should be left aligned. The first number should be an integer in hexadecimal; the second should be fractional positive; and the third – a negative fraction. The last two numbers have to be rounded to the second decimal place.
5. Write a program that reads from the console two integer numbers (`int`) and prints how many numbers between them exist, such that the remainder of their division by 5 is 0. Example: in the range (14, 25) there are 3 such numbers: 15, 20 and 25.



- 
6. Write a program that reads two numbers from the console and prints the greater of them. Solve the problem without using conditional statements.
  7. Write a program that reads five integer numbers and prints their sum. If an invalid number is entered the program should prompt the user to enter another number.
  8. Write a program that reads five numbers from the console and prints the greatest of them.
  9. Write a program that reads an integer number  $n$  from the console. After that reads  $n$  numbers from the console and prints their sum.
  10. Write a program that reads an integer number  $n$  from the console and prints all numbers in the range  $[1...n]$ , each on a separate line.
  11. Write a program that prints on the console the first 100 numbers in the Fibonacci sequence: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, ...
  12. Write a program that calculates the sum (with precision of 0.001) of the following sequence:  $1 + 1/2 - 1/3 + 1/4 - 1/5 +$
- 

# Comparison Operators

Operator	Action
<code>==</code>	Equal to
<code>!=</code>	Not equal to
<code>&gt;</code>	Greater than
<code>&gt;=</code>	Greater than or equal to
<code>&lt;</code>	Less than
<code>&lt;=</code>	Less than or equal to

# Comparison Operators

```
// Comparison operators
int weight = 700;
Console.WriteLine(weight >= 500); // True
char gender = 'm';
Console.WriteLine(gender <= 'f'); // False
double colorWaveLength = 1.630;
Console.WriteLine(colorWaveLength > 1.621); // True
int a = 5;
int b = 7;
bool condition = (b > a) && (a + b < a * b);
Console.WriteLine(condition); // True
Console.WriteLine('B' == 'A' + 1); // True
```

# Comparison of Integers and Characters

```
// Comparison of Integers and Characters
Console.WriteLine("char 'a' == 'a'? " + ('a' == 'a')); // True
Console.WriteLine("char 'a' == 'b'? " + ('a' == 'b')); // False
Console.WriteLine("5 != 6? " + (5 != 6)); // True
Console.WriteLine("5.0 == 5L? " + (5.0 == 5L)); // True
Console.WriteLine("true == false? " + (true == false)); // False
```

# Comparison of References to Objects

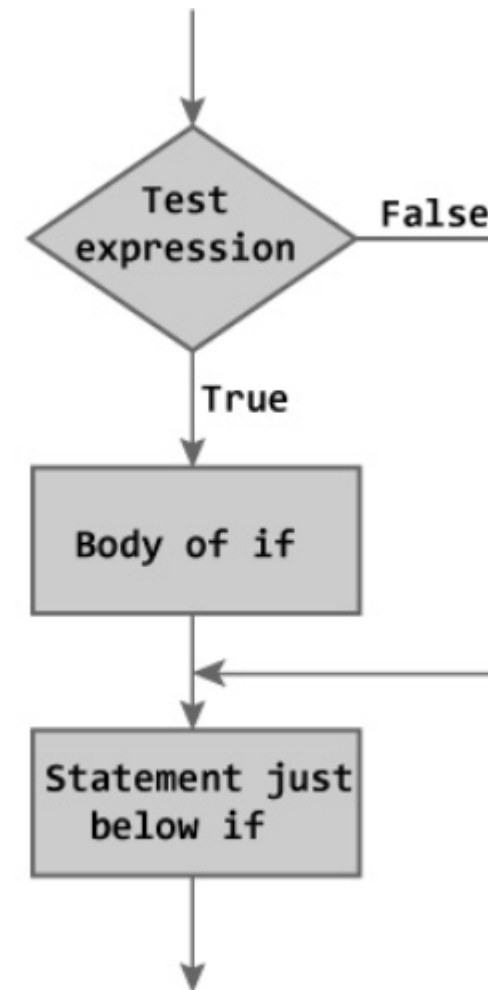
```
// Comparison of References to Objects
string str = "beer";
string anotherStr = str;
string thirdStr = "bee";
thirdStr = thirdStr + 'r';
Console.WriteLine("str = {0}", str);
Console.WriteLine("anotherStr = {0}", anotherStr);
Console.WriteLine("thirdStr = {0}", thirdStr);
Console.WriteLine(str == anotherStr); // True - same object
Console.WriteLine(str == thirdStr); // True - equal objects
Console.WriteLine((object)str == (object)anotherStr); // True
Console.WriteLine((object)str == (object)thirdStr); // False
```

## Logical Operators

```
// Logical Operators
bool result1 = (2 < 3) && (3 < 4);    // True
bool result2 = (2 < 3) || (1 == 2);  // True
Console.WriteLine("Exclusive OR: " + ((2 < 3) ^ (4 > 3)));
// Exclusive OR: False
bool value = !(7 == 5); // True
Console.WriteLine(value);
```

# If Statement

```
if (Boolean expression)
{
    Body of the conditional statement;
}
```



# If Statement example

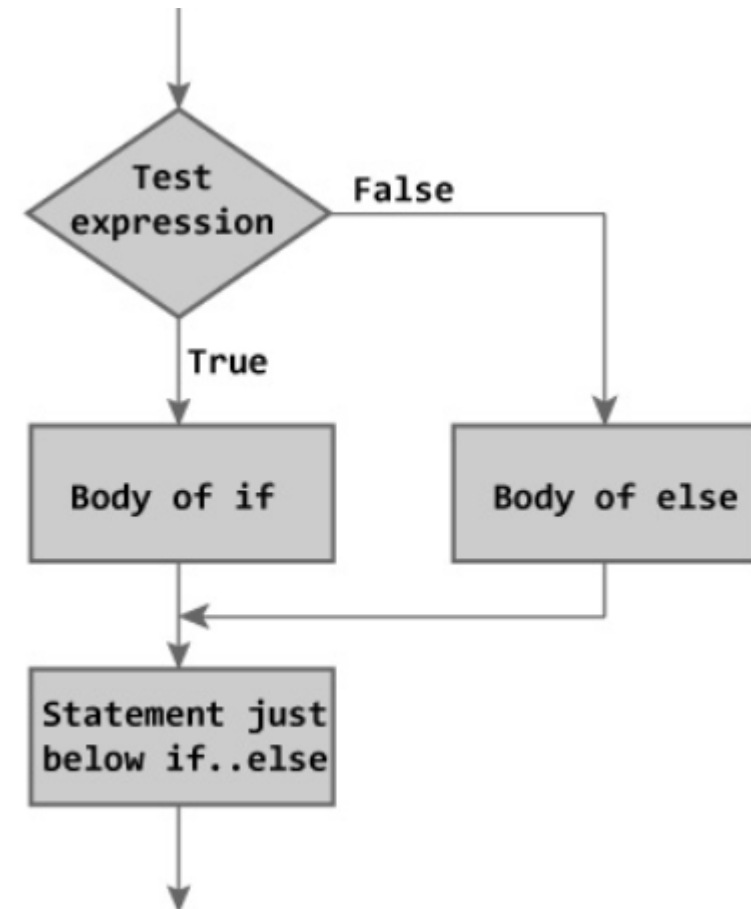
```
Console.WriteLine("Enter two numbers.");
Console.Write("Enter first number: ");
int firstNumber = int.Parse(Console.ReadLine());
Console.Write("Enter second number: ");
int secondNumber = int.Parse(Console.ReadLine());
int biggerNumber = firstNumber;
if (secondNumber > firstNumber)
{
    biggerNumber = secondNumber;
}
Console.WriteLine("The bigger number is: {0}", biggerNumber);
//Enter two numbers.
//Enter first number: 4
//Enter second number: 5
//The bigger number is: 5

int a = 6;
if (a > 5)
    Console.WriteLine("The variable is greater than 5.");
    Console.WriteLine("This code will always execute!");
// Bad practice: misleading code
```



# If else statement

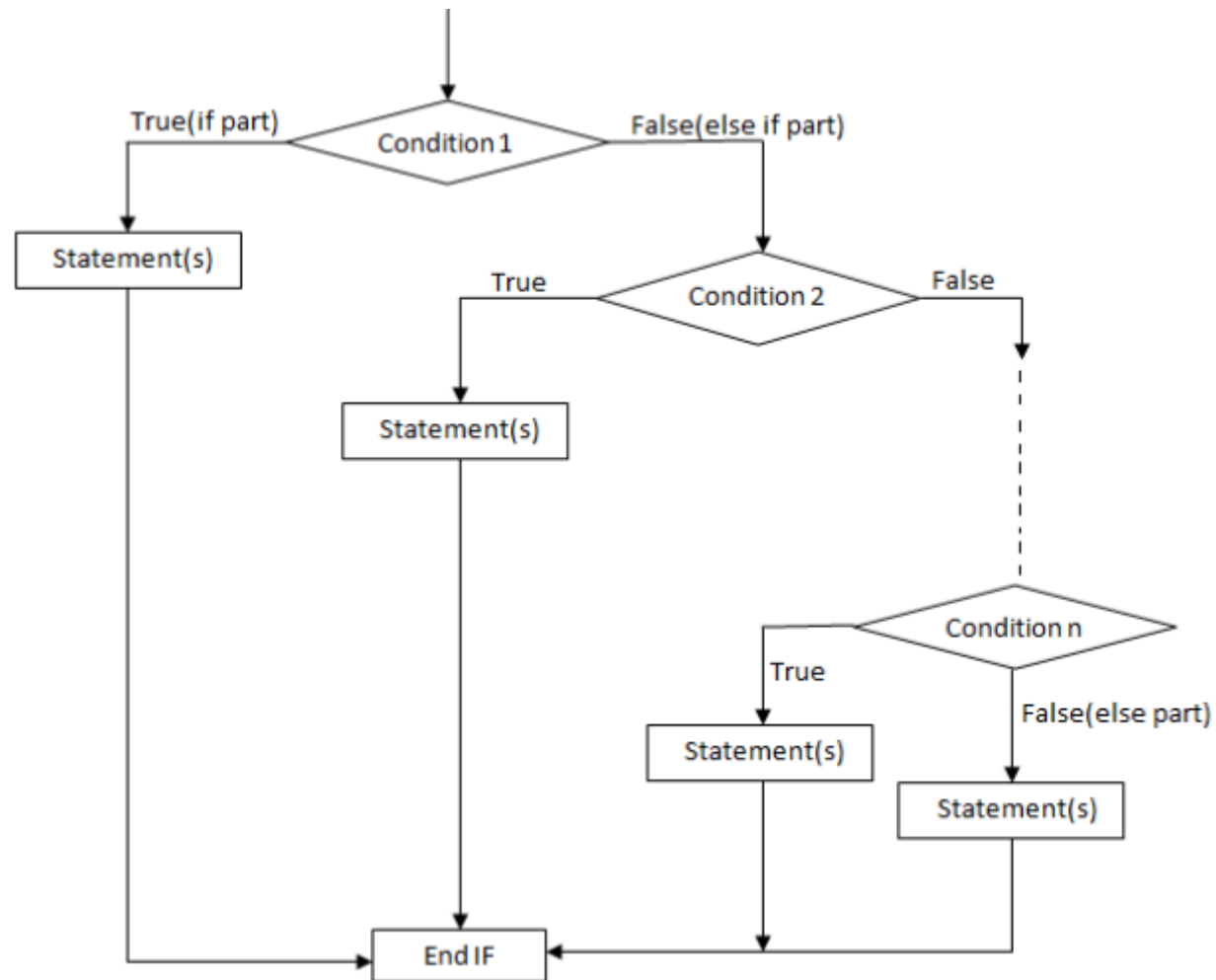
```
if (Boolean expression)
{
    Body of the conditional statement;
}
else
{
    Body of the else statement;
}
```



# If else example

```
// if else example
int x = 2;
if (x > 3)
{
    Console.WriteLine("x is greater than 3");
}
else
{
    Console.WriteLine("x is not greater than 3");
}
// x is not greater than 3
```

# Nested " if " Statements



# Nested " if " Statements – Example

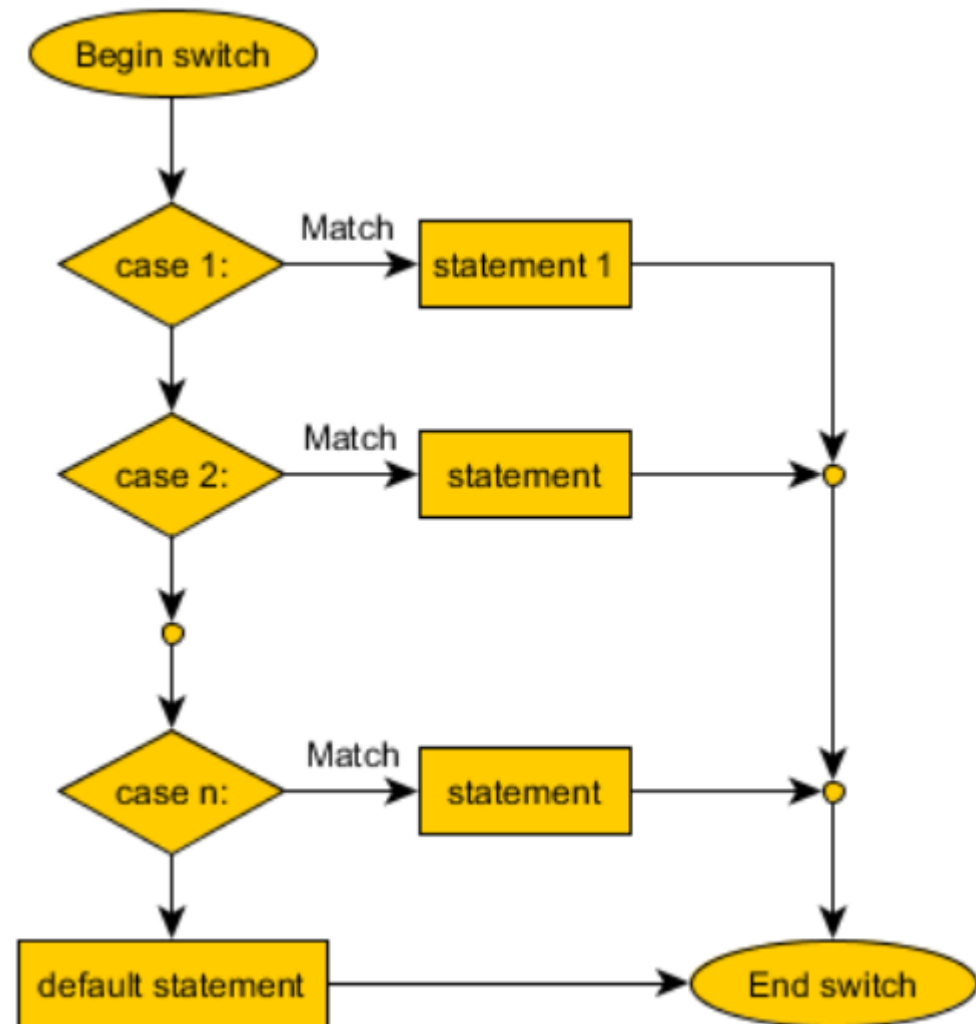
```
// Nested "if" Statements - Example
int first = 5;
int second = 3;
if (first == second)
{
    Console.WriteLine("These two numbers are equal.");
}
else
{
    if (first > second)
    {
        Console.WriteLine("The first number is greater.");
    }
    else
    {
        Console.WriteLine("The second number is greater.");
    }
}
// The first number is greater.
```

# If else if statement

```
char ch = 'X';
if (ch == 'A' || ch == 'a')
{
    Console.WriteLine("Vowel [ei]");
}
else if (ch == 'E' || ch == 'e')
{
    Console.WriteLine("Vowel [i:]");
}
else if (ch == 'I' || ch == 'i')
{
    Console.WriteLine("Vowel [ai]");
}
else if (ch == 'O' || ch == 'o')
{
    Console.WriteLine("Vowel [ou]");
}
else if (ch == 'U' || ch == 'u')
{
    Console.WriteLine("Vowel [ju:]");
}
else
{
    Console.WriteLine("Consonant");
}
// Consonant
```

# Switch-case statement

```
switch (integer_selector)
{
    case integer_value_1:
        statements;
        break;
    case integer_value_2:
        statements;
        break;
    // ...
    default:
        statements;
        break;
}
```



# switch-case example

```
int number = 6;
switch (number)
{
    case 1:
    case 4:
    case 6:
    case 8:
    case 10:
        Console.WriteLine("The number is not prime!"); break;
    case 2:
    case 3:
    case 5:
    case 7:
        Console.WriteLine("The number is prime!"); break;
    default:
        Console.WriteLine("Unknown number!"); break;
}
// The number is not prime!
```

# Exercises

1. Write an if-statement that takes two integer variables and exchanges their values if the first one is greater than the second one.
2. Write a program that shows the sign (+ or -) of the product of three real numbers, without calculating it. Use a sequence of if operators.
3. Write a program that finds the biggest of three integers, using nested if statements.
4. Sort 3 real numbers in descending order. Use nested if statements.
5. Write a program that asks for a digit (0-9), and depending on the input, shows the digit as a word (in English). Use a switch statement.
6. Write a program that gets the coefficients  $a$ ,  $b$  and  $c$  of a quadratic equation:  $ax^2 + bx + c$ , calculates and prints its real roots (if they exist). Quadratic equations may have 0, 1 or 2 real roots.
7. Write a program that finds the greatest of given 5 numbers.



---

8. Write a program that, depending on the user's choice, inputs int, double or string variable. If the variable is int or double, the program increases it by 1. If the variable is a string, the program appends "\*" at the end. Print the result at the console. Use switch statement.

9. We are given 5 integer numbers. Write a program that finds those subsets whose sum is 0. Examples:

- If we are given the numbers {3, -2, 1, 1, 8}, the sum of -2, 1 and 1 is 0.
- If we are given the numbers {3, 1, -7, 35, 22}, there are no subsets with sum 0.

10. Write a program that applies bonus points to given scores in the range [1...9] by the following rules:

- If the score is between 1 and 3, the program multiplies it by 10.
- If the score is between 4 and 6, the program multiplies it by 100.
- If the score is between 7 and 9, the program multiplies it by 1000.
- If the score is 0 or more than 9, the program prints an error message.

---

11. \* Write a program that converts a number in the range [0...999] to words, corresponding to the English pronunciation. Examples:

- 0 --> "Zero"
- 12 --> "Twelve"
- 98 --> "Ninety eight"
- 273 --> "Two hundred seventy three"
- 400 --> "Four hundred"
- 501 --> "Five hundred and one"
- 711 --> "Seven hundred and eleven"