

# Console Input / Output

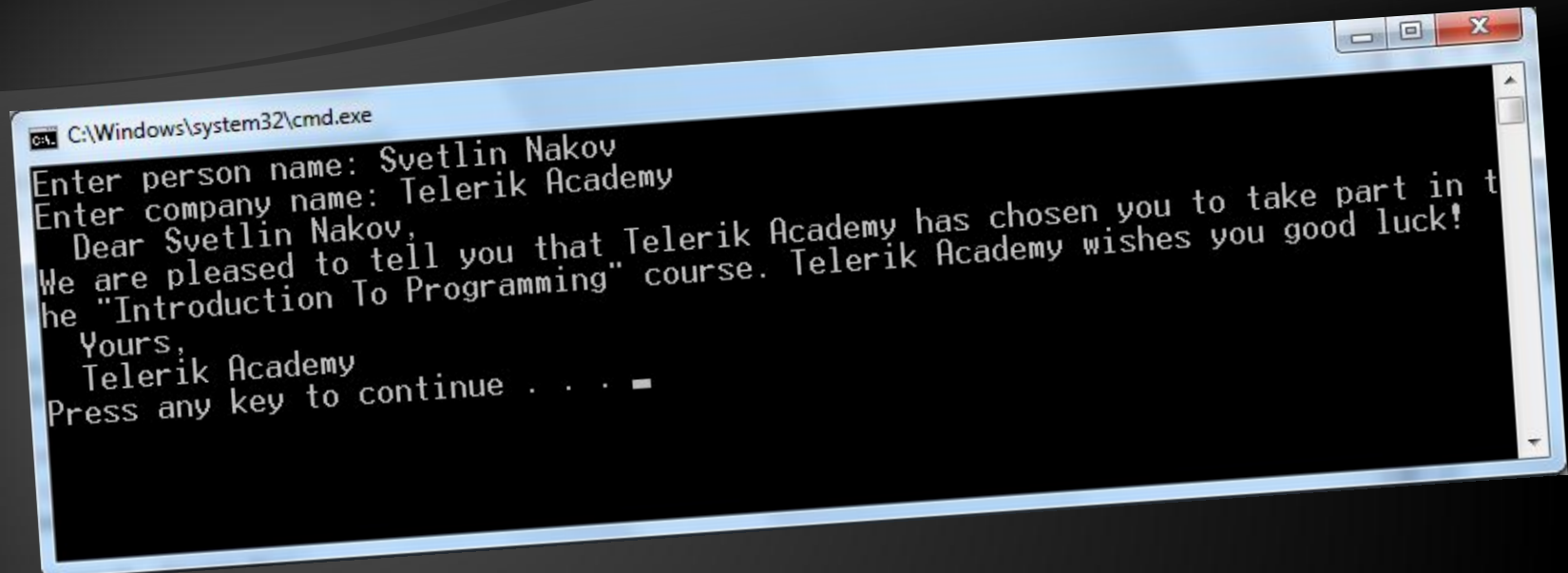
Reading and Writing to the Console

---



# Table of Contents

- ◆ **Printing to the Console**
  - ◆ **Printing Strings and Numbers**
- ◆ **Reading from the Console**
  - ◆ **Reading Characters**
  - ◆ **Reading Strings**
  - ◆ **Parsing Strings to Numeral Types**
  - ◆ **Reading Numeral Types**
- ◆ **Various Examples**



```
C:\Windows\system32\cmd.exe
Enter person name: Svetlin Nakov
Enter company name: Telerik Academy
Dear Svetlin Nakov,
We are pleased to tell you that Telerik Academy has chosen you to take part in t
he "Introduction To Programming" course. Telerik Academy wishes you good luck!
Yours,
Telerik Academy
Press any key to continue . . . -
```

# Printing to the Console

Printing Strings, Numeral Types and Expressions

# Printing to the Console

- ◆ Console is used to display information in a text window
- ◆ Can display different values:
  - ◆ Strings
  - ◆ Numeral types
  - ◆ All primitive data types
- ◆ To print to the console use the class `Console` (`System.Console`)

# The Console Class

- ◆ Provides methods for console input and output
  - ◆ Input
    - ◆ `Read(...)` – reads a single character
    - ◆ `ReadKey(...)` – reads a combination of keys
    - ◆ `ReadLine(...)` – reads a single line of characters
  - ◆ Output
    - ◆ `Write(...)` – prints the specified argument on the console
    - ◆ `WriteLine(...)` – prints specified data to the console and moves to the next line

# Console.WriteLine(...)

- ◆ Printing an integer variable

```
int a = 15;  
...  
Console.WriteLine(a); // 15
```

- ◆ Printing more than one variable using a formatting string

```
double a = 15.5;  
int b = 14;  
...  
Console.WriteLine("{0} + {1} = {2}", a, b, a + b);  
// 15.5 + 14 = 29.5
```

- ◆ Next print operation will start from the same line

# Console.WriteLine(...)

- ◆ Printing a string variable

```
string str = "Hello C#!";  
...  
Console.WriteLine(str);
```

- ◆ Printing more than one variable using a formatting string

```
string name = "Marry";  
int year = 1987;  
...  
Console.WriteLine("{0} was born in {1}.", name, year);  
// Marry was born in 1987.
```

- ◆ Next printing will start from the next line

# Printing to the Console – Example

```
static void Main()
{
    string name = "Peter";
    int age = 18;
    string town = "Sofia";

    Console.Write("{0} is {1} years old from {2}.",
        name, age, town);
    // Result: Peter is 18 years old from Sofia.
    Console.Write("This is on the same line!");
    Console.WriteLine("Next sentence will be" +
        " on a new line.");

    Console.WriteLine("Bye, bye, {0} from {1}.",
        name, town);
}
```



# Using Parameters – Example

```
static void Main()
{
    int a=2, b=3;
    Console.Write("{0} + {1} =", a, b);
    Console.WriteLine(" {0}", a+b);
    // 2 + 3 = 5

    Console.WriteLine("{0} * {1} = {2}",
        a, b, a*b);
    // 2 * 3 = 6

    float pi = 3.14159206;
    Console.WriteLine("{0:F2}", pi); // 3,14

    Console.WriteLine("Bye - Bye!");
}
```

# Printing a Menu – Example

```
double colaPrice = 1.20;
string cola = "Coca Cola";
double fantaPrice = 1.20;
string fanta = "Fanta Dizzy";
double zagorkaPrice = 1.50;
string zagorka = "Zagorka";

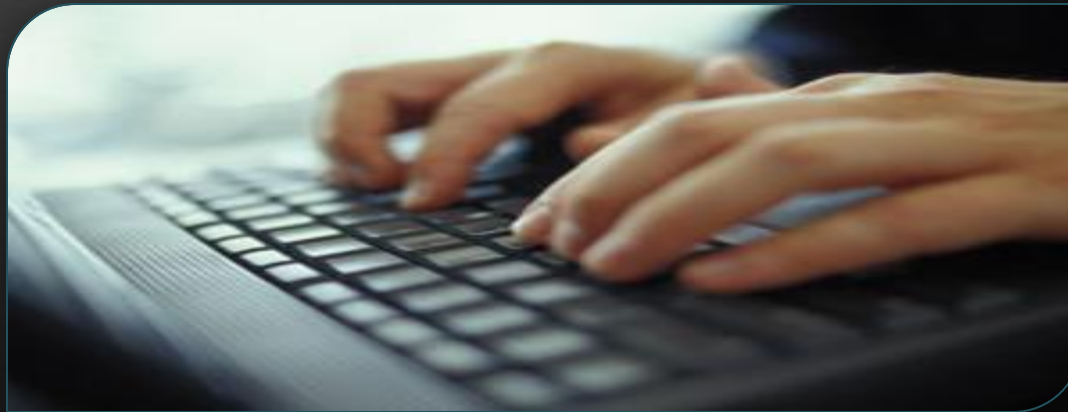
Console.WriteLine("Menu:");
Console.WriteLine("1. {0} - {1}",
    cola, colaPrice);
Console.WriteLine("2. {0} - {1}",
    fanta, fantaPrice);
Console.WriteLine("3. {0} - {1}",
    zagorka, zagorkaPrice);
Console.WriteLine("Have a nice day!");
```

# Printing to the Console

Live Demo

# Reading from the Console

Reading Strings and Numeral Types



# Reading from the Console

- ◆ We use the console to read information from the command line
- ◆ We can read:
  - ◆ Characters
  - ◆ Strings
  - ◆ Numeral types (after conversion)
- ◆ To read from the console we use the methods `Console.Read()` and `Console.ReadLine()`

# Console.Read()

- ◆ Gets a single character from the console (after [Enter] is pressed)
  - ◆ Returns a result of type `int`
  - ◆ Returns `-1` if there aren't more symbols
- ◆ To get the actually read character we need to cast it to `char`

```
int i = Console.Read();  
char ch = (char) i; // Cast the int to char  
  
// Gets the code of the entered symbol  
Console.WriteLine("The code of '{0}' is {1}.", ch, i);
```

# Reading Characters from the Console

Live Demo

# Console.ReadKey()

- ◆ Waits until a combination of keys is pressed
  - ◆ Reads a single character from console or a combination of keys
- ◆ Returns a result of type `ConsoleKeyInfo`
  - ◆ `KeyChar` – holds the entered character
  - ◆ `Modifiers` – holds the state of [Ctrl], [Alt], ...

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



# Reading Keys from the Console

Live Demo

# Console.ReadLine()

- ◆ Gets a line of characters
- ◆ Returns a string value
- ◆ Returns null if the end of the input is reached

```
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);
```

# Reading Strings from the Console

Live Demo

# Reading Numeral Types

- ◆ Numeral types can not be read directly from the console
- ◆ To read a numeral type do the following:
  1. Read a string value
  2. Convert (parse) it to the required numeral type
- ◆ `int.Parse(string)` – parses a string to `int`

```
string str = Console.ReadLine()  
int number = int.Parse(str);  
  
Console.WriteLine("You entered: {0}", number);
```

# Converting Strings to Numbers

- ◆ Numeral types have a method `Parse(...)` for extracting the numeral value from a string
  - ◆ `int.Parse(string)` – `string`  $\square$  `int`
  - ◆ `long.Parse(string)` – `string`  $\square$  `long`
  - ◆ `float.Parse(string)` – `string`  $\square$  `float`
  - ◆ Causes `FormatException` in case of error

```
string s = "123";  
int i = int.Parse(s); // i = 123  
long l = long.Parse(s); // l = 123L
```

```
string invalid = "xxx1845";  
int value = int.Parse(invalid); // FormatException
```

# Reading Numbers from the Console – Example

```
static void Main()
{
    int a = int.Parse(Console.ReadLine());
    int b = int.Parse(Console.ReadLine());

    Console.WriteLine("{0} + {1} = {2}",
        a, b, a+b);
    Console.WriteLine("{0} * {1} = {2}",
        a, b, a*b);

    float f = float.Parse(Console.ReadLine());
    Console.WriteLine("{0} * {1} / {2} = {3}",
        a, b, f, a*b/f);
}
```

# Converting Strings to Numbers (2)

- ◆ Converting can also be done using the methods of the Convert class
  - ◆ `Convert.ToInt32(string)` – `string`  $\square$  `int`
  - ◆ `Convert.ToSingle(string)` – `string`  $\square$  `float`
  - ◆ `Convert.ToInt64(string)` – `string`  $\square$  `long`
  - ◆ Internally uses the parse methods of the numeral types

```
string s = "123";  
int i = Convert.ToInt32(s); // i = 123  
long l = Convert.ToInt64(s); // l = 123L  
  
string invalid = "xxx1845";  
int value = Convert.ToInt32(invalid); // FormatException
```

# Reading Numbers from the Console

Live Demo



# Error Handling when Parsing

- ◆ Sometimes we want to handle the errors when parsing a number
  - ◆ Two options: use try-catch block or TryParse()
- ◆ Parsing with TryParse():

```
string str = Console.ReadLine();
int number;
if (int.TryParse(str, out number))
{
    Console.WriteLine("Valid number: {0}", number);
}
else
{
    Console.WriteLine("Invalid number: {0}", str);
}
```

# Parsing with TryParse()

Live Demo

# Reading and Printing to the Console

## Various Examples



# Printing a Letter – Example

```
Console.Write("Enter person name: ");
string person = Console.ReadLine();

Console.Write("Enter company name: ");
string company = Console.ReadLine();

Console.WriteLine(" Dear {0},", person);
Console.WriteLine("We are pleased to tell you " +
    "that {1} has chosen you to take part " +
    "in the \"Introduction To Programming\" " +
    "course. {1} wishes you good luck!",
    person, company);

Console.WriteLine(" Yours,");
Console.WriteLine(" {0}", company);
```



# Printing a Letter

Live Demo

# Calculating Area – Example

```
Console.WriteLine("This program calculates " +  
    "the area of a rectangle or a triangle");  
  
Console.Write("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.Write("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 {0}", area);
```

# Calculating Area

Live Demo



# Summary

- ◆ We have discussed the basic input and output methods of the class `Console`
  - ◆ `Write(...)` and `WriteLine(...)`
    - ◆ Used to write values to the console
  - ◆ `Read(...)` and `ReadLine(...)`
    - ◆ Used to read values from the console
- ◆ Parsing numbers to strings
  - ◆ `int.Parse(...)`, `double.Parse(...)`, ...



# Exercises

1. Write a program that reads 3 integer numbers from the console and prints their sum.
2. Write a program that reads the radius  $r$  of a circle and prints its perimeter and area.
3. A company has name, address, phone number, fax number, web site and manager. The manager has first name, last name, age and a phone number. Write a program that reads the information about a company and its manager and prints them on the console.

## Exercises (2)

4. Write a program that reads two positive integer numbers and prints how many numbers  $p$  exist between them such that the remainder of the division by 5 is 0 (inclusive). Example:  $p(17, 25) = 2$ .
5. Write a program that gets two numbers from the console and prints the greater of them. Don't use `if` statements.
6. Write a program that reads the coefficients  $a$ ,  $b$  and  $c$  of a quadratic equation  $ax^2+bx+c=0$  and solves it (prints its real roots).

## Exercises (3)

7. Write a program that gets a number  $n$  and after that gets more  $n$  numbers and calculates and prints their sum.
8. Write a program that reads an integer number  $n$  from the console and prints all the numbers in the interval  $[1..n]$ , each on a single line.
9. Write a program to print the first 100 members of the sequence of Fibonacci: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, ...
10. Write a program to calculate the sum (with accuracy of 0.001):  $1 + 1/2 - 1/3 + 1/4 - 1/5 + \dots$