

2015



C#/ .NET
programming basics

The logo features a stylized profile of a human head in grey, with a yellow brain inside. The brain is depicted with white lines representing neural connections.

BASIC PRINCIPLES OF C#, CLR

BRAIN
ACADEMY

Training program

- **Block 1.** C# programming fundamentals
- **Block 2.** Windows application development
- **Block 3.** Service-oriented and web application development
- **Block 4.** Application architecture and design patterns
- **Block 5.** Certification

Block 1 contents

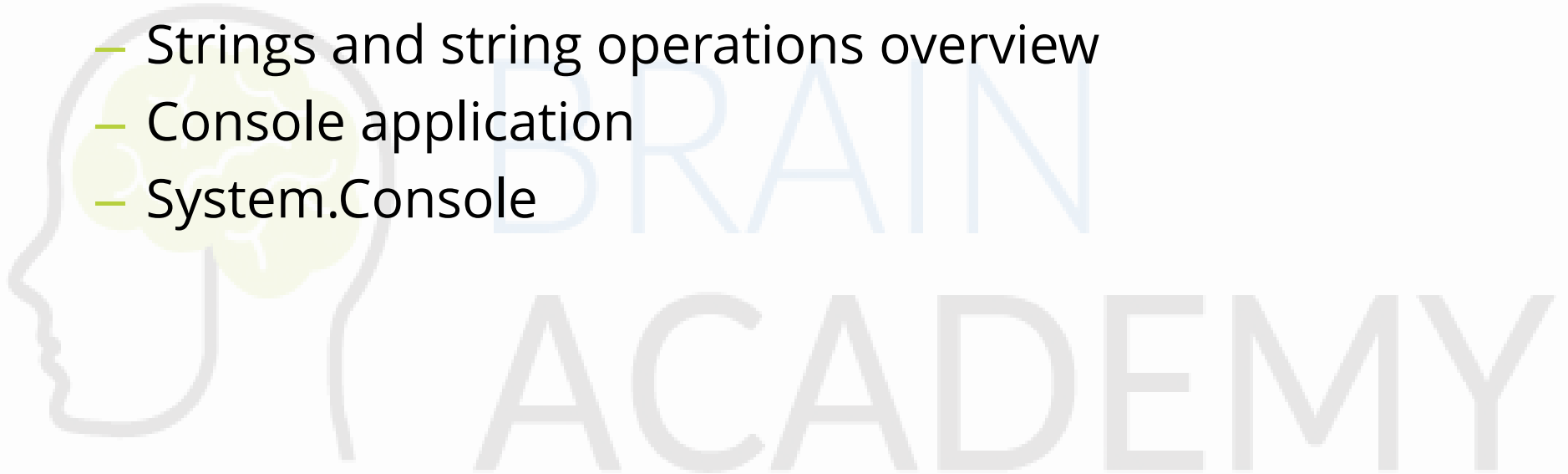
1. Basic principles of C#, CLR
2. Object oriented programming fundamentals
3. Exception handling
4. Advanced programming (Delegates, events, lambdas. Generics. Collections)
5. Assembly management and application debug
6. Multithreading and asynchronous processing
7. Data access
8. Unsafe code and pointers. .Net Framework security

System.Console

- Basic principles of C#, CLR
 - C# & CLR basics
 - Data types
 - Operators
 - Array, Structure, Enum
 - System.Console

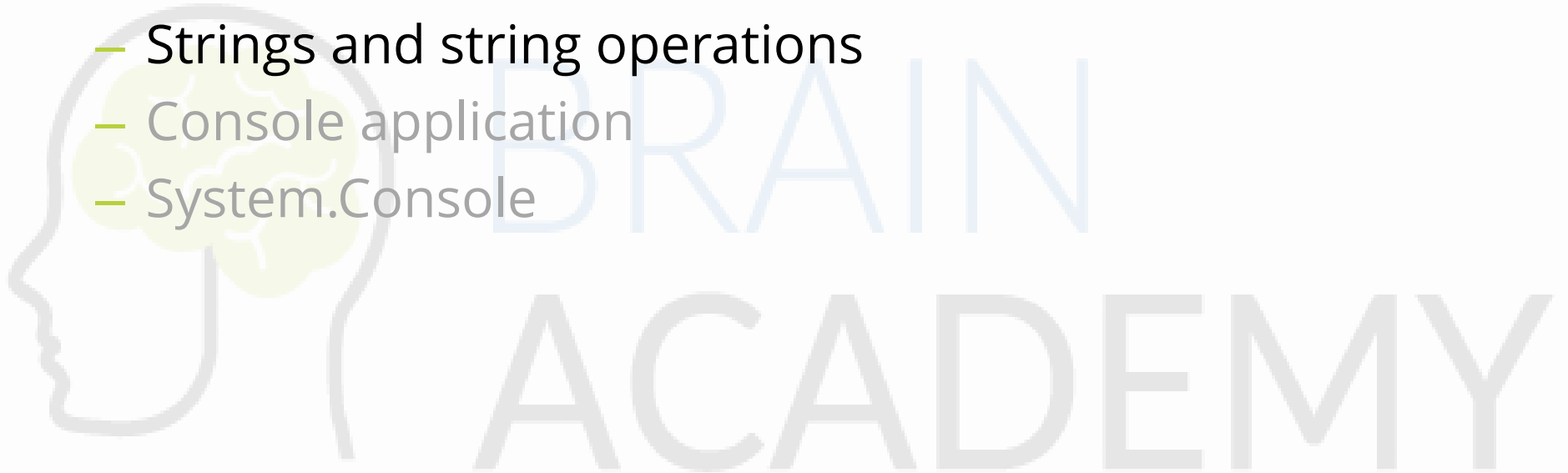
Lecture contents

- System.Console
 - Strings and string operations overview
 - Console application
 - System.Console



Strings and string operations overview

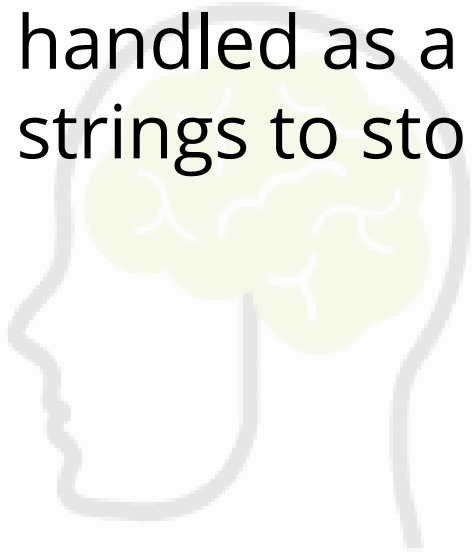
- System.Console
 - Strings and string operations
 - Console application
 - System.Console



Strings



String - a group of characters or character bytes handled as a single entity. Computer programs use strings to store and transmit data and commands.



BRAIN
ACADEMY

String type (1/2)

- **string** is a type (as **int**, **double**, **char**)
 - `string my_str = "some string";`
- **string** is **reference** type
- **string** represents a sequence of characters
 - `char [] chars = {'s', 'o', 'm', 'e', ' ', 's', 't', 'r', 'i', 'n', 'g'};`
 - `string my_str = new string(chars);`

String type (2/2)

- **string** is alias of **System.String**
 - `System.String my_str = "some string";`
- **string** is **immutable** type
 - `my_str[0] = 'S'; // compilation error`

String literals

- Can contain any character literal
 - Literal: `"some string"`
 - `my_str = "some string \n";`
- Verbatim string literal start with symbol `@`
 - `@ "C:\MyFolder\MyTextFile.txt"`
 - `// equals to`
 - `// "C:\MyFolder\MyTextFile.txt"`

String operations overview

- Concatenation

- `my_str = "some " + "string";`
 - `my_str += "!!!";`

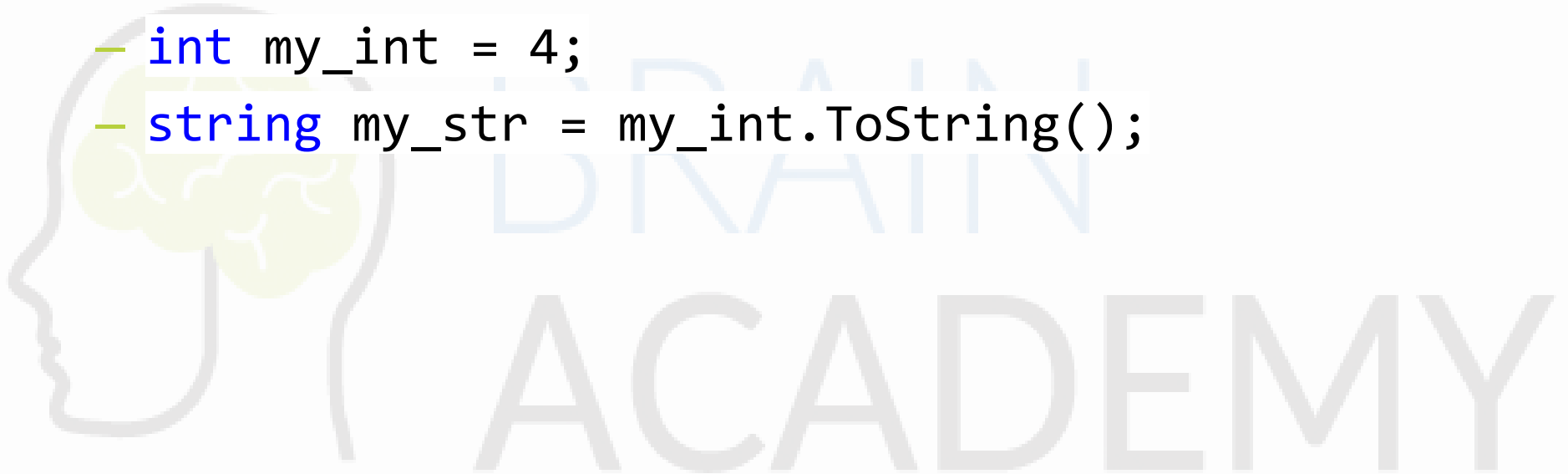
- [] operator

- `char ch = my_str[0];`

ToString()

- **ToString()** – method for casting to string type

- `int my_int = 4;`
 - `string my_str = my_int.ToString();`



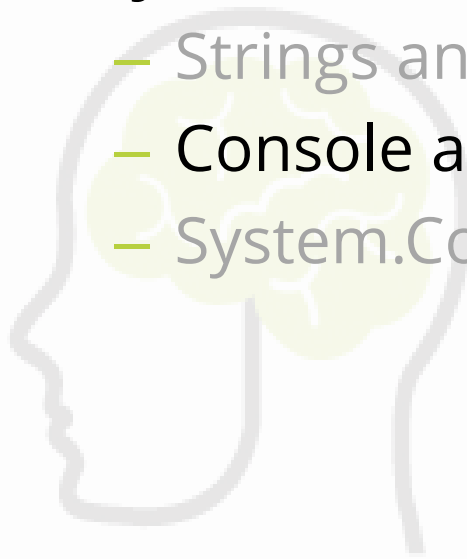
Converting from string

- **System.Convert**

- `string my_str = "425";`
 - `int my_int = Convert.ToInt32(my_str);`

Console application

- System.Console
 - Strings and string operations overview
 - Console application
 - System.Console

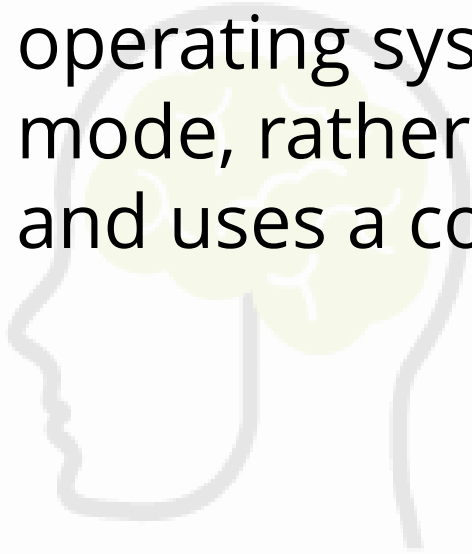


BRAIN
ACADEMY

Console application (1/2)



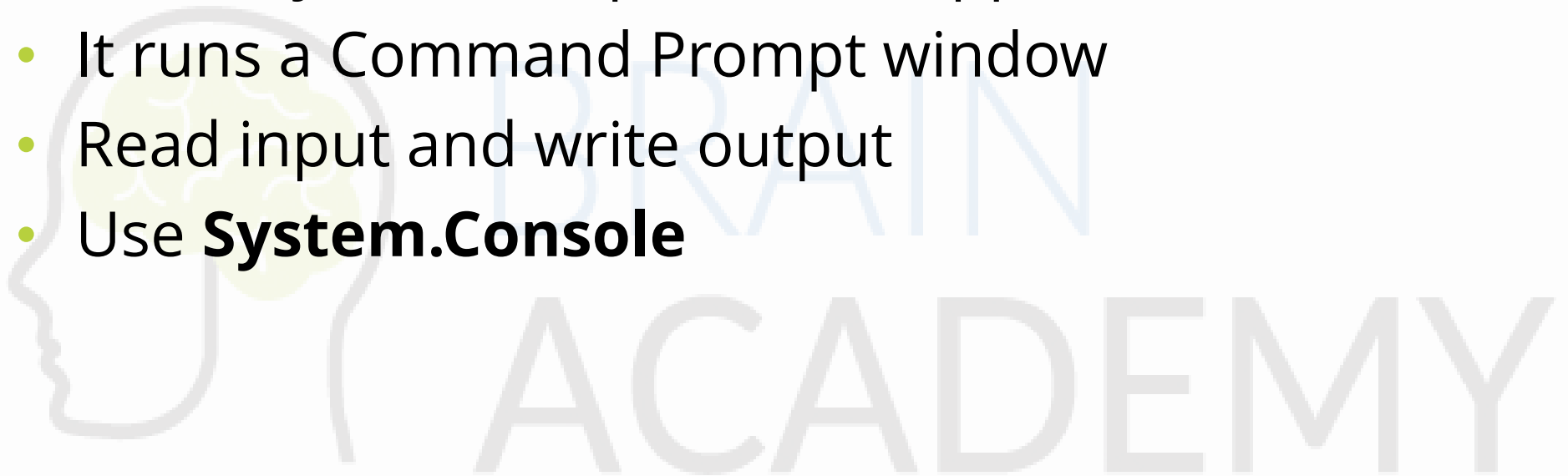
Console application – a program that runs from the operating system's command line, in character-mode, rather than from a graphical user interface and uses a console window for its input and output.



BRAIN
ACADEMY

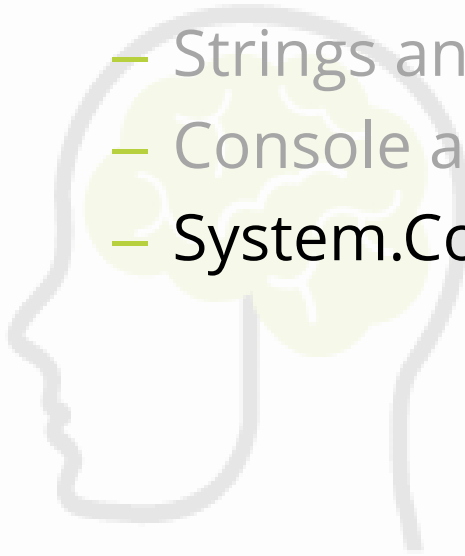
Console application (2/2)

- It is easy to develop Console Application
- It runs a Command Prompt window
- Read input and write output
- Use **System.Console**



System.Console

- System.Console
 - Strings and string operations overview
 - Console application
 - System.Console



BRAIN
ACADEMY

System.Console (1/2)

- Include standard input, output, error streams for **Console Application**
- **Write(), WriteLine()** – output
- **Read(), ReadLine(), ReadKey()** – input
- **Beep()** – plays sound
- **Clear()** – clear console buffer

System.Console (2/2)

- **BackgroundColor** - Gets or sets the background color of the console
- **ForegroundColor** - Gets or sets the foreground color of the console.
- **BufferHeight/ BufferWidth** - Gets or sets the height/ width of the buffer area
- **Error** - Gets the standard error output stream
- **In/ Out** - Gets the standard input/ output stream
- **WindowHeight/ WindiwWidth** - Gets or sets the height/ width of the console window area



System.Console example

```
1. Console.Write("Hello ");
2. Console.WriteLine("World!");
3. Console.Write("Your name: ");
4. string name = Console.ReadLine();
5. Console.Write("Hi, ");
6. Console.Write(name);
7. Console.WriteLine("!");
8. Console.ReadKey();
```