COMP 1202
Object Oriented Programming
Lab Session - Week 04 - Loops

Mehdi Azad

January 30, 2024

Organization

Instructor

- mehdi.azad@georgebrown.ca

TA

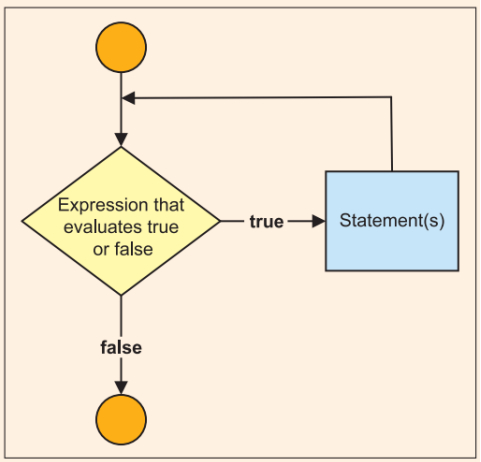
- sherry.zarei@georgebrown.ca

Course Code

- Github: https://github.com/mabbasiazad/COMP1202

While Loop

```
while (conditional expression)
      statement(s);
```

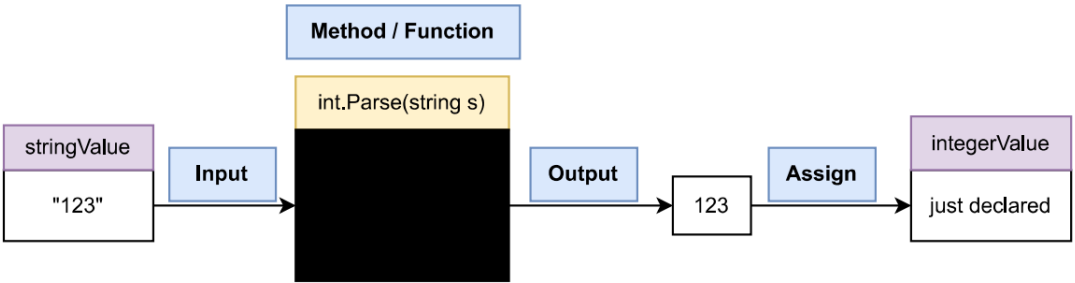


While Loop Example

Example: compute the following statement: sum of numbers from 0 to N

$\sum_{k=0}^{N} k = 0 + 1 + 2 + ... + N$

Parse() Vs. TryParse() (1)




```
string stringValue = "123"
int intergerValue;
integerValue = int.Parse(stringValue);
```

Interpreting the code

```
string stringValue = "123"
int intergerValue;
integerValue = int.Parse(stringValue);
```

- By writing int.Parse(stringValue), I **call** the **method** *int.Parse()* and **pass** *stringValue* as its **input** ("123").
- The **method** takes this **input** and process it and finally gives me an **output** (123)
- I get this **output** and **assign** it to *integerValue*

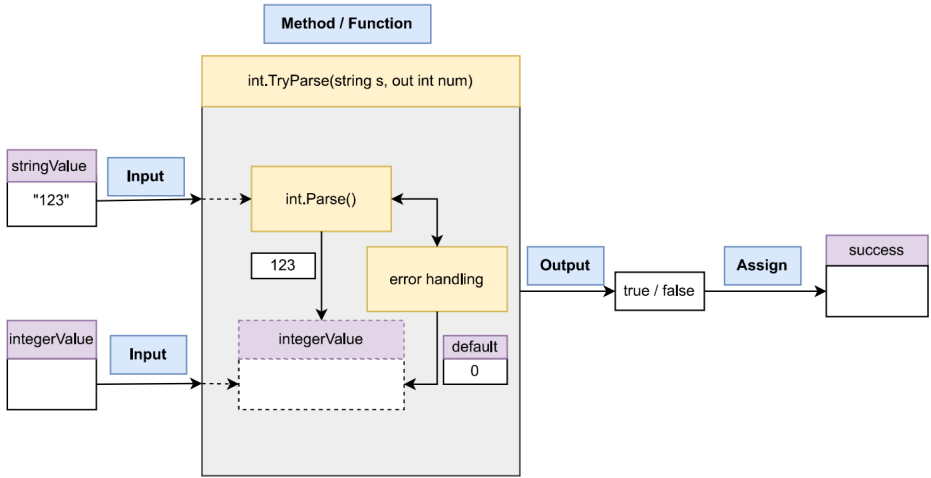
Interpreting the code

integerValue = int.Parse(stringValue)

can be simplified as:
integerValue = int.Parse("123")

can be simplified as:
integerValue = 123

Parse() Vs. TryParse() (2)




```
bool success;
sucess = int.TryParse(stringValue, out integerValue);
```

What is bool type?

- **bool** is a data type like *int*, *double*, *string* that you've already seen.
- The value of a bool type can be True or False.
- For example the expression $(3 > 5)$ is false. if you run this program you'll see that false is stored in the value *b*.

```
public static void Main(string[] args)
{
    bool b; //this is bool type which can be
       true or false
    b = (3 > 5);
    Console.WriteLine($"the amount stored in b
       is: {b}");
}
```

TryParse()

```
string inValue;
inValue = Console.ReadLine();
int number;
bool success = int.TryParse(inValue, out number);
if (success == true)
{
    Console.WriteLine($"Converted '{inValue}' to
       {number} was success");
}
else
{
    Console.WriteLine("Invalid input - 0 recorded
       for number");
}
```

Valid Input

1- write a program to get an integer input from the user. If the user enters invalid number the program should detect that and ask the user to re-enter the input until the input is valid.

helper: use int.TryParse and while loop

2- Change your program is a way that it just accepts number between 0 to 100.

Lab Exercise 3 / 7

Write a program that gets two double numbers *a* and *b* and compute *a / b*. If *b == 0* then *a / b* is not defined. So for getting *b* ask the user to enter their number until *b != 0*.

Note: use double.Parse(); no need to use double.TryParse()

Solution name	week04
Project name	DivByZero

neamespace	week04
class	Divide
method	Main()

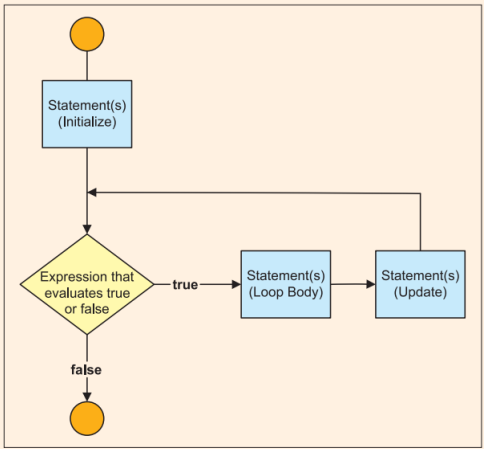
Name	Type	Description
inputA	string	user first input
inputB	string	user second input
a	double	inputA converted to double
b	double	inputB converted to double
result	double	a / b

Calculator Application

- Read the calculator app program. Run it and try to understand the code
- change it in such a way that it can handle division by zero
- change the program in a way that it just accept clean input
- (optional) change the program in a way that it can do infinite number of operation. But there is special code 999; If the user enters this code for the first number the calculator program stops. (helper: put the whole program in Main() method inside while(true) and break the loop if num1 = 999)

For Loop

```
for (initialize; test; update)
      statement;
```



For Loop Example

create this sequence:
$0, 1^2, 2^2, 3^2, 4^2, ..., N^2$

hint: use C# builtin method called Math.Pow(a, b) to calculate a^b

Writing a simple Method

Write temperature conversion from Celsius to Fahrenheit using a method