For Loop

Softuni Team

Technical Trainers

SoftUni

Table of Contents

1. **Review** of the Previous Lesson
2. **Increment** and **Decrement** Operators
   - Prefix and Postfix **++** and **--**
3. **For Loops**: Repeating Blocks of Code
4. For Loop with a **Step**
5. Iterating over **Characters**
6. **Infinite** Loops

Have a Question?

SoftUni

sli.do

#prgm-for-qa

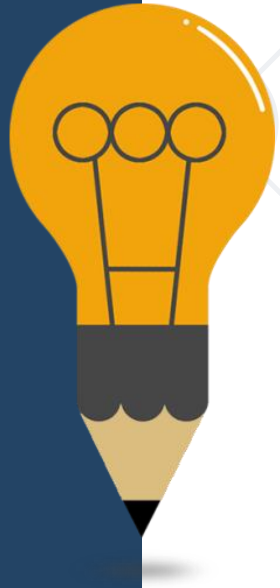
Review

Conditional Statements Advanced

Nested Conditions

- An **if…else** statement can be **nested** within another **if…else** statement
  - Test one condition, followed by another

```
if (expression) {
  if (nested_expression)
    // Some code for execution
  else
    // Other code for execution
}
```



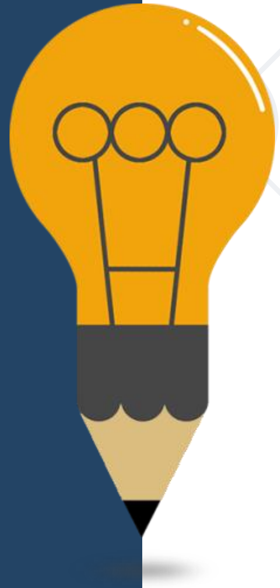
Conditional Operators

- **Logical operators** (such as **AND**, **OR**, **NOT**) are used to build complex logical conditions
- The logical operators in Java are:
  - AND – **&&**
  - OR – **||**
  - Logical negation – **!**
  - Brackets – **( )**

Switch-Case

- Choosing among a list of possibilities
- Alternative to an `if-else` statement

```
switch (selector) {
  case someCase:
    statements;
    break;
  default:
    statements;
    break;
}
```



Increment and Decrement

Using **++** and --

Increment / Decrement Operators

- Increment (**++**) operator **increases** the value **by 1**
- Decrement (**--**) operator **decreases** the value **by 1**
- Can be used **prefix** and **postfix** form
  - Prefix: `++i`, `--i`
  - Postfix: `i++`, `i--`
- Both operators can be used only with numeric variables

Example: Increment

- **Prefix** increment

```
int a = 1;
Console.WriteLine(++a); // 2
Console.WriteLine(a);   // 2
```

Increases the value and then prints it

- **Postfix** increment

```
int a = 1;
Console.WriteLine(a++); // 1
Console.WriteLine(a);   // 2
```

First prints the value and then increases it

Example: Decrement

- **Prefix** decrement

```
int a = 1;
Console.WriteLine(--a); // 0
Console.WriteLine(a);   // 0
```

Decreases the value and then prints it

- **Postfix** decrement

```
int a = 1;
Console.WriteLine(a--); // 1
Console.WriteLine(a);   // 0
```

First prints the value and then decreases it

Loops: Introduction

For-Loops

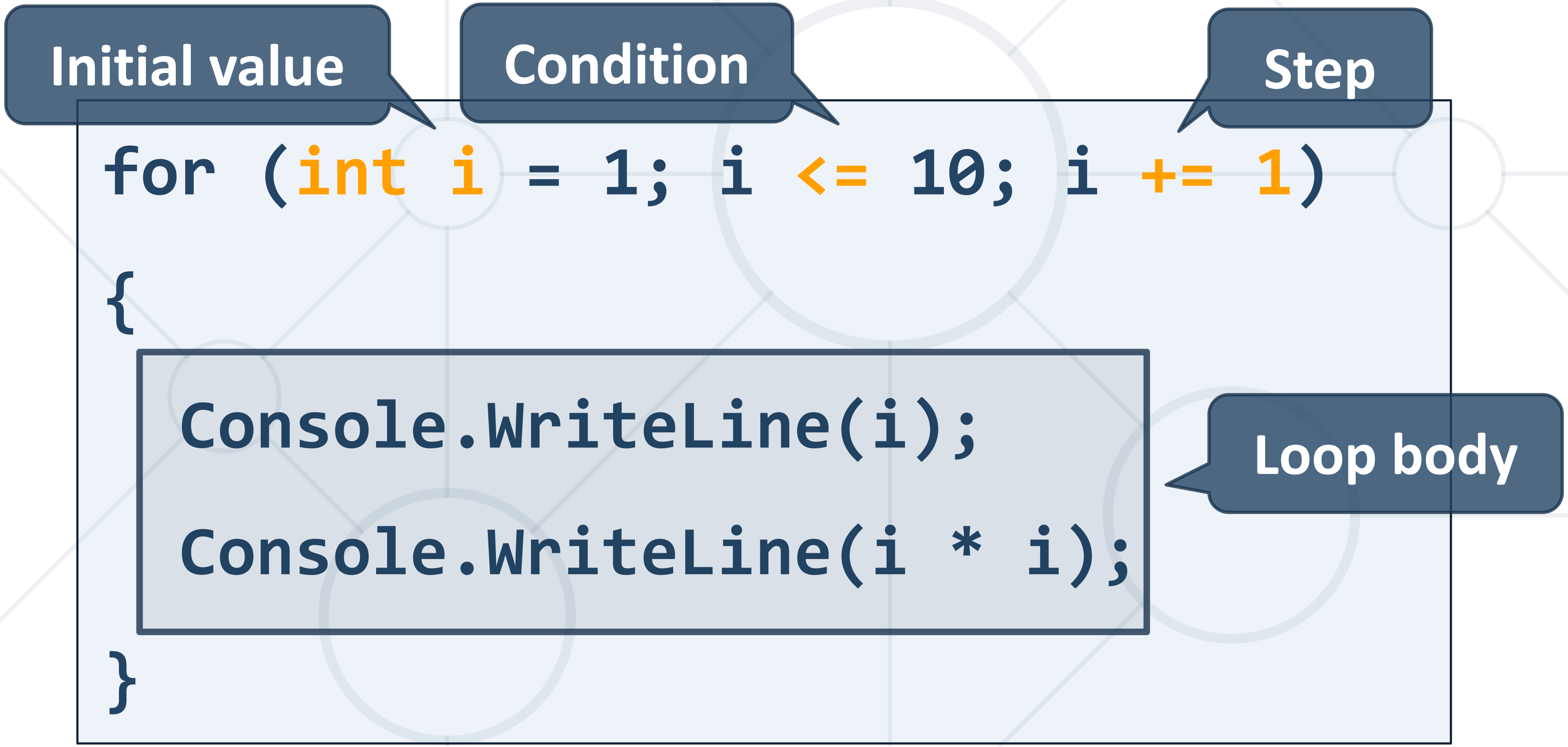
For-Loop Example: Dishes

- Filling the dishwasher machine

For-Loop

Control Flow Statement

For-Loop: Example



Initial value

Condition

Step

```
for (int i = 1; i <= 10; i += 1)
{
  Console.WriteLine(i);
  Console.WriteLine(i * i);
}
```

Loop body

For-Loop

- Allows code to be executed **repeatedly**
  - While certain **condition** is true

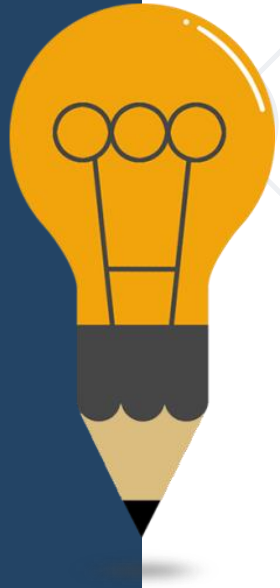
```
for (initialization; condition; step)
{
    // Body of the for loop
}
```

- **Initialization** – initializes the loop variable
- **Condition** – logical exit condition
- **Step** – updates the loop variable

For-Loop – Examples

- Print the **numbers 1 … 20**:

```
for (int i = 1; i <= 20; i++)
    Console.WriteLine(i);
```



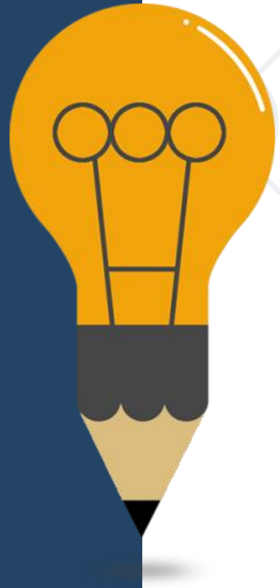
- Print the **numbers 100 … 200**:

```
for (int i = 100; i <= 200; i++)
    Console.WriteLine(i);
```

For-Loop – More Examples

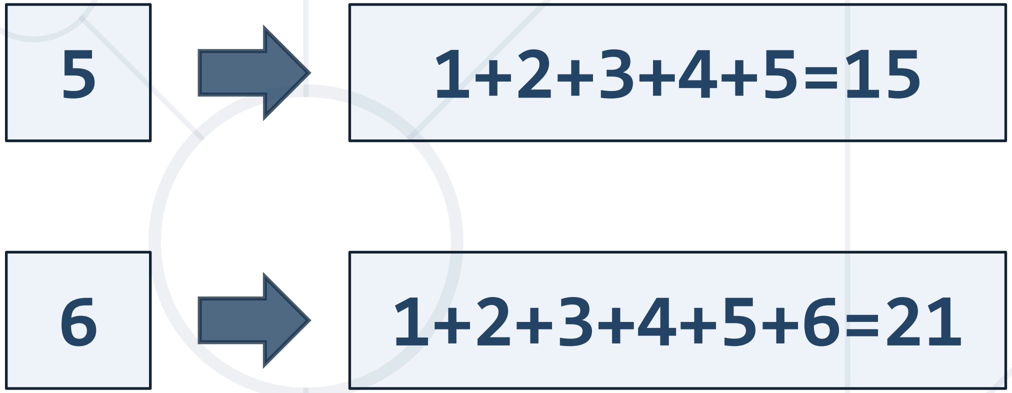
- Print the **numbers 1 … 20** and their **square**

```
for (int x = 1; x <= 20; x += 1)
{
  int square = x * x;
  Console.WriteLine($"{x} * {x} = {square}");
}
```



Problem: First N Numbers Sum

- Write a program, which **sums the numbers 1…n**:
  - Reads number **n** from the console
  - Sums all numbers from **1** to **n**
  - Prints the **sum** on the console as shown below:



```
5 → 1+2+3+4+5=15

6 → 1+2+3+4+5+6=21
```

Solution: Print Sum of N Numbers

```
int n = int.Parse(Console.ReadLine());
int sum = 1;
Console.Write(1);
for (int i = 2; i <= n; i += 1)
{
 Console.Write("+" + i);
  sum += i;
}
Console.WriteLine("=" + sum);
```

Problem: Sum N Numbers

- Write a program to **sum given N numbers**:
  - Read **n** – the count of numbers to sum
  - Read **n floating-point numbers** and print their **sum**

```
3
10
20
30
```

```
60
```

```
4
2.5
3.5
0.3
0.9
```

```
7.2
```

Solution: Sum N Numbers

```
int n = int.Parse(Console.ReadLine());
double sum = 0;
for (int i = 0; i < n; i += 1) {
  sum += double.Parse(Console.ReadLine());
}
Console.WriteLine(sum);
```

Loops with a Step

Positive and Negative Loop Step

For Loop with Step

SoftUni

- The **step** part in a for loop can either **increase** or **decrease** the value of a variable, even with a **step**

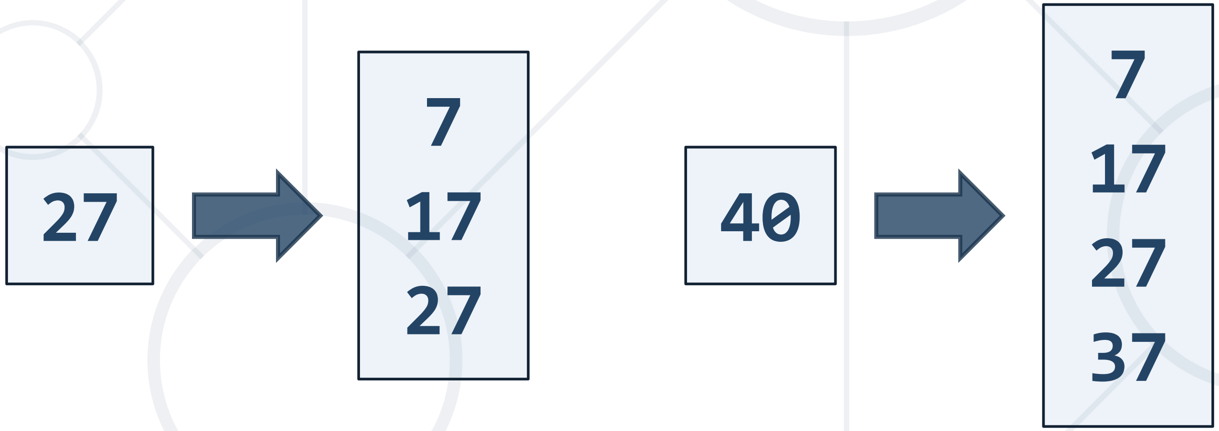
```
for (int i = 0; i < 10; i += 2)
  Console.WriteLine(i);
```

```
for (int i = 10; i >= 0; i -= 2)
  Console.WriteLine(i);
```

Always pay attention on the condition

Problem: Numbers Ending with 7

- Write a program to print **numbers ending in 7** in given range:
  - Reads a number **n**
  - Prints all numbers from **7** to **n**, ending with 7



```
27
```

```
7
17
27
```

```
40
```

```
7
17
27
37
```

Solution: Numbers Ending with 7

```
int n = int.Parse(Console.ReadLine());
for (int i = 7; i <= n; i += 10)
{
  Console.WriteLine(i);
}
```

Problem: Exam Countdown

- Write a program to print a **countdown to an exam** (see below):
  - Read an integer **d**: the count of days before an exam
  - For each day **d…1** print: "**{currentDay} days before the exam**"
  - At the end print: "**`The exam has come`**"

```
3
```

→

```
3 days before the exam
2 days before the exam
1 days before the exam
The exam has come
```

Solution: Exam Countdown

```csharp
int days = int.Parse(Console.ReadLine());
for (int i = days; i >= 1; i -= 1)
{
  Console.WriteLine($"{i} days before the exam", i);
}
Console.WriteLine("The exam has come");
```

Iterating over Characters

The ASCII Table

- Computers can only understand numbers
- **ASCII** code is the numerical representation of a character

Decimal	Hex	Html	Char
97	61	&#97;	a
98	62	&#98;	b

- **Unicode** is more powerful character encoding standard: https://techterms.com/definition/unicode

  - '**a**' has the int value (ASCII code) of **97**
  - '**b**' has the int value (ASCII code) of **98**
  - Learn more at: https://ascii-code.com

Iterating over Characters

- In C#, we can **iterate over characters**

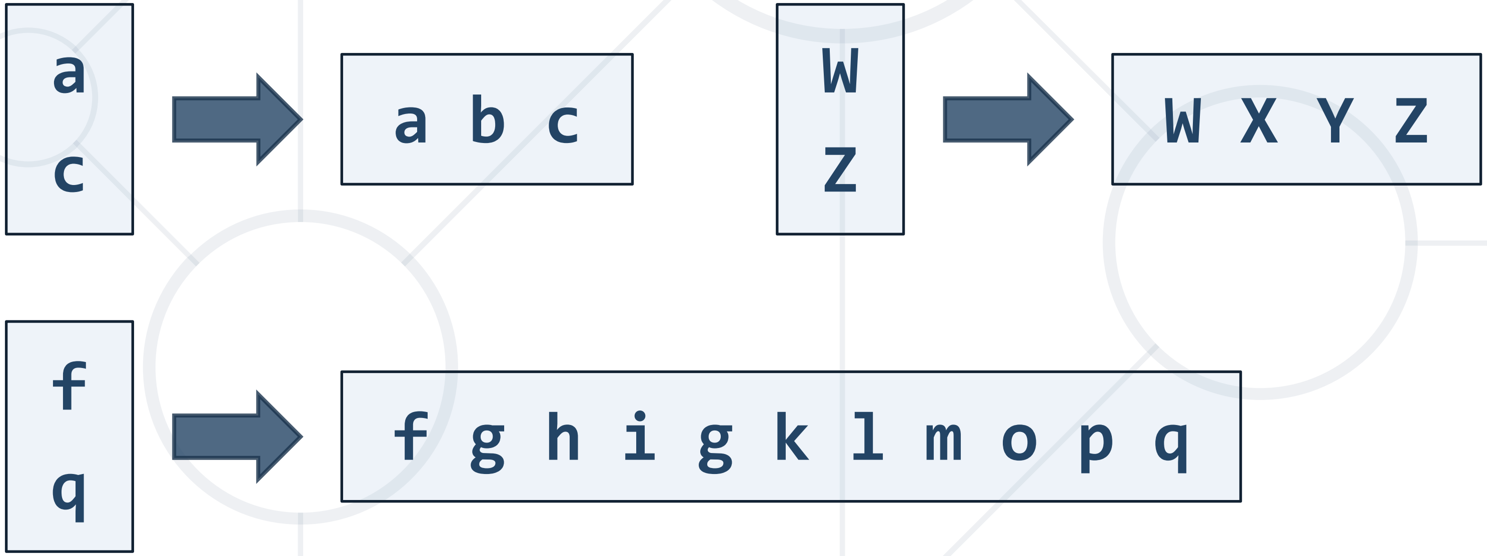
```
for (char ch = 'a'; ch <= 'f'; ch++)
{
  Console.Write(ch + " ");
}
```

- Convert **ASCII** / **Unicode** number to **char**:

```
char ch = (char) 65;
Console.WriteLine(ch); // A
```

Problem: Latin Letters

- Write a program to print the **Latin letters in certain range**:
  - Read **2 letters**, each on separate line
  - Print all letters in the specified range **inclusively**



```
a
c
```

```
a b c
```

```
W
Z
```

```
W X Y Z
```

```
f
q
```

```
f g h i g k l m o p q
```

Solution: Latin Letters

```
char startLetter = char.Parse(Console.ReadLine());
char endLetter = char.Parse(Console.ReadLine());
for (char i = startLetter; i <= endLetter; i++)
{
  Console.Write(i + " ");
}
```

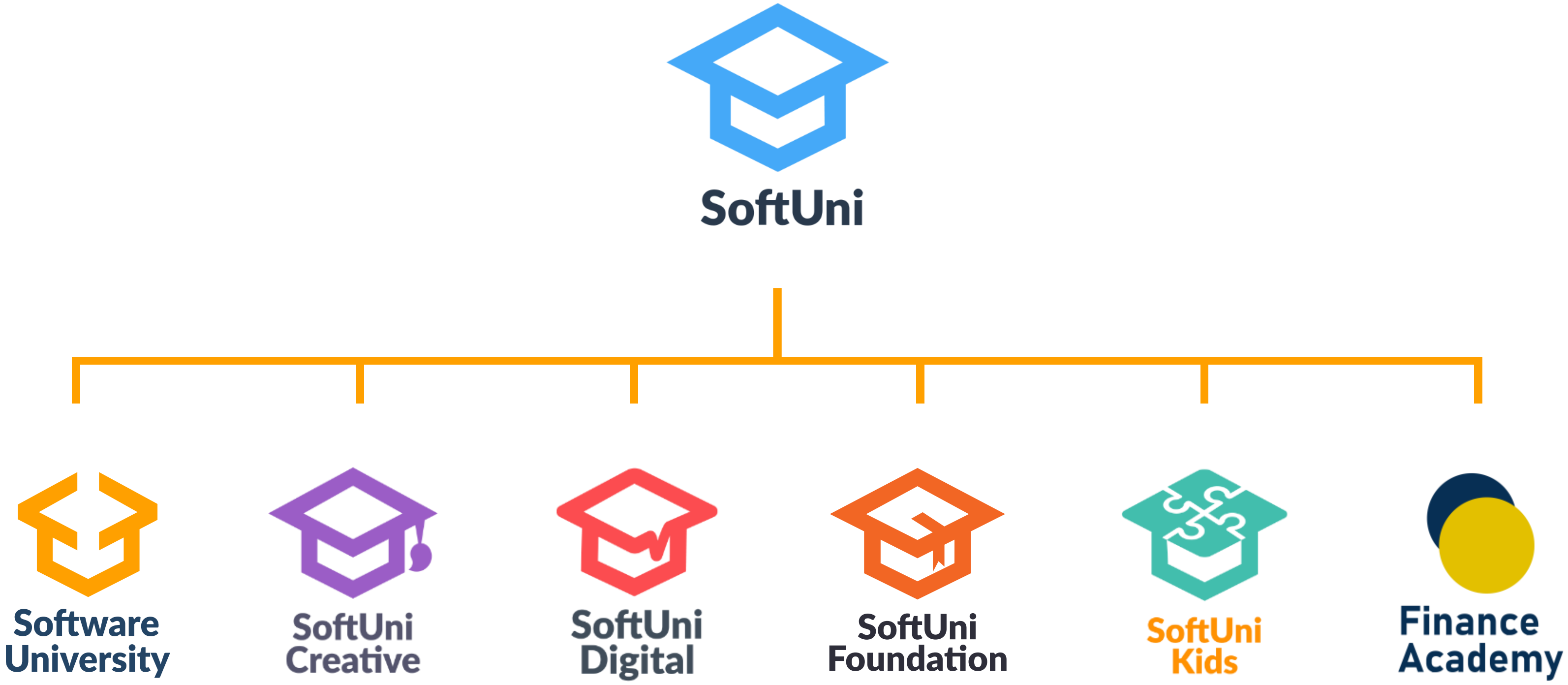
Summary: For-Loops

- For loops execute a block of code multiple times
- For-loop components:
  - Initialization
  - Condition
  - Step
  - Body

```
for (int i = 0; i < 9; i++)
{
  Console.WriteLine(i);
}
```

Software University

Questions?



SoftUni

Software University

SoftUni Creative

SoftUni Digital

SoftUni Foundation

SoftUni Kids

Finance Academy

SoftUni Diamond Partners

SoftUni

SUPER HOSTING .BG

Coca-Cola HBC Bulgaria

Flutter™ International

INDEAVR
Serving the high achievers

AMBITIONED

DRAFT KINGS

SOFTWARE GROUP

BOSCH

Postbank
Решения за твоето утре

PHAR VISION

SmartIT

DXC TECHNOLOGY

createX

Trainings @ Software University (SoftUni)

- Software University – High-Quality Education, Profession and Job for Software Developers
  - softuni.bg, about.softuni.bg
- Software University Foundation
  - softuni.foundation
- Software University @ Facebook
  - facebook.com/SoftwareUniversity
- Software University Forums
  - forum.softuni.bg

Software
University

FORUM

License