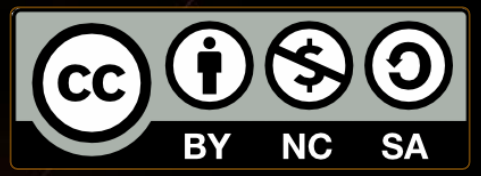
Introduction to C Programming

The C Language, History, Writing Your First Program in C



SoftUni Team

Technical Trainers

Software University

http://softuni.bg

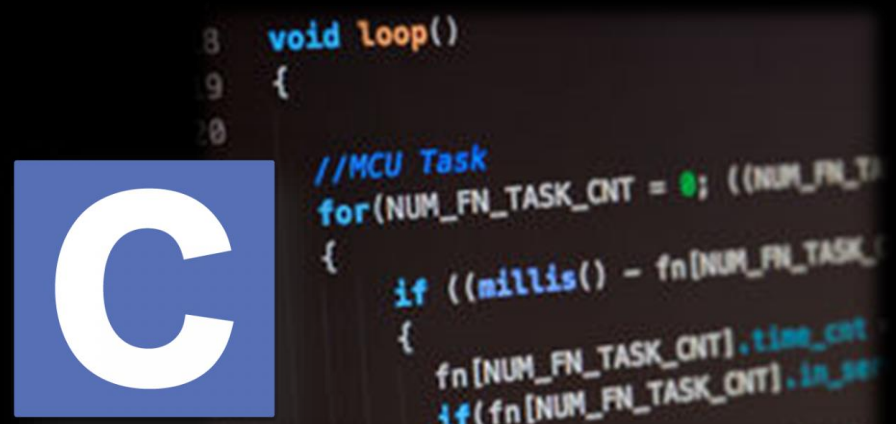


Table of Contents

SOFTWARE UNIVERSITY FOUNDATION

What is Computer Programming?

Computer Programming

Computer programming: creating a sequence of instructions to enable the computer to do something

Definition by Google

Software Development Phases

- Define a task / problem = **Specification**
- Plan your solution = **Architecture / Design**
  - Find suitable algorithm / data structures to use
  - Find suitable libraries / platforms / frameworks
- Write source code (step by step) = **Implementation**
- Fix program errors (bugs) = **Testing & Debugging**
- Install, configure and run the software = **Deployment**
- Fix / improve the software over time = **Maintenance**

```
int i = 0;
for(i = 0; i < NUM_TESTS; i++) {
    Stack_push(stack, tests[i]);
    mu_assert(Stack_peek(stack) == tests[i], "Wrong next valu
}

mu_assert(Stack_count(stack) == NUM_TESTS, "Wrong count on pu

STACK_FOREACH(stack, cur) {
    debug("VAL: %s", (char *)cur->value);
}

for(i = NUM_TESTS - 1; i >= 0; i--) {
    char *val = Stack_pop(stack);
```

```
#include <stdio.h>
```

The C Programming Language

The C Language – History

- The C language was developed by Dennis Ritchie at Bell Labs
  - First released in 1972 (over 40 years ago)
  - Compiles directly to binary
  - Has very high performance
  - Can run on almost any hardware
  - Has a standard library with many built-in functions
  - Widely used to develop performance-demanding systems
    - E.g. operating systems, embedded systems, device drivers, etc.

Your First C Program

Your First C Program

SOFTWARE UNIVERSITY FOUNDATION

Include the C standard Input/Output Library

```
#include <stdio.h>

int main()
{
    printf("Hello, C!\n");

    return 0;
}
```

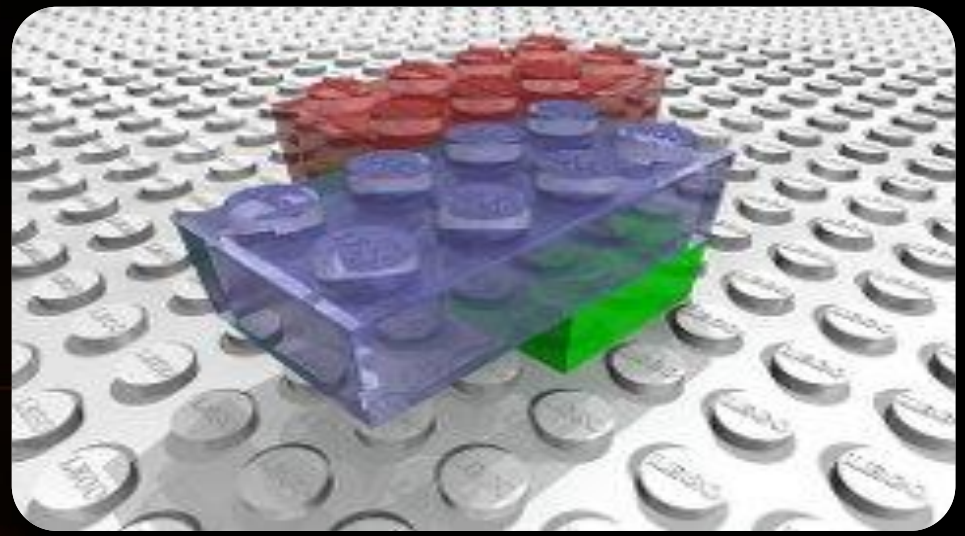
Define a function called **main**

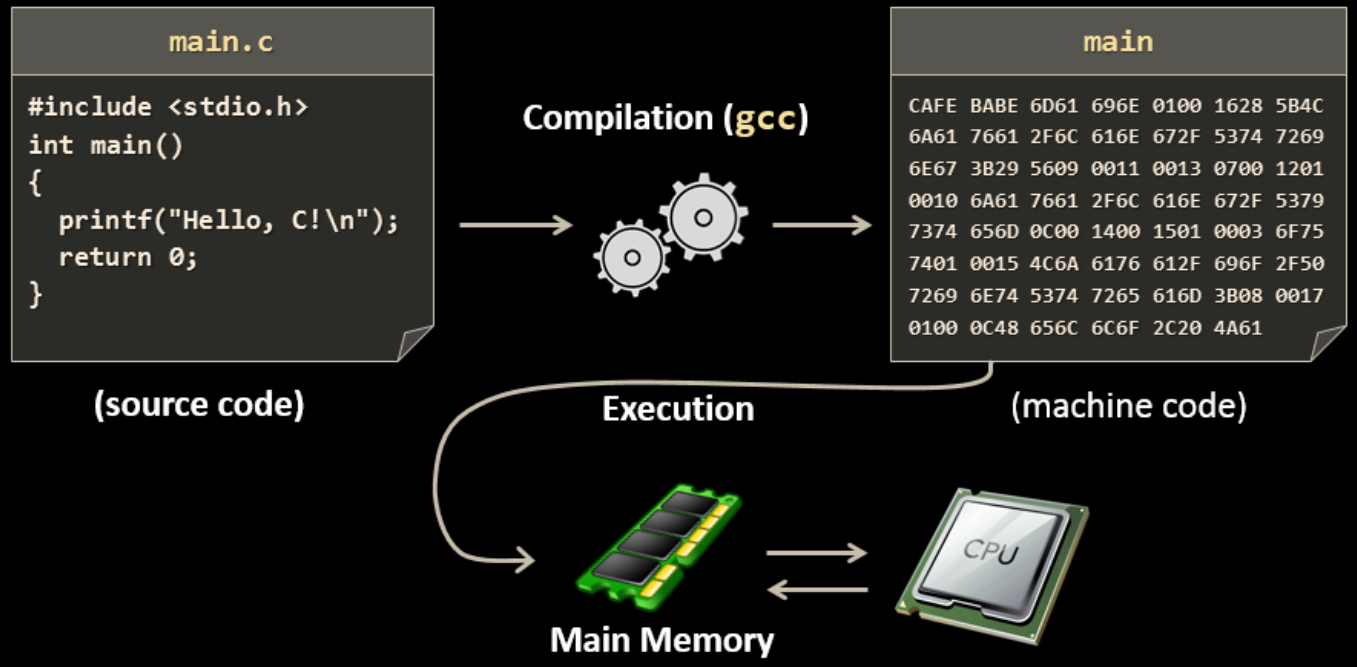
Call **printf()** function to print on the console

Exit program by returning **0**

Your First C# Program

Live Demo





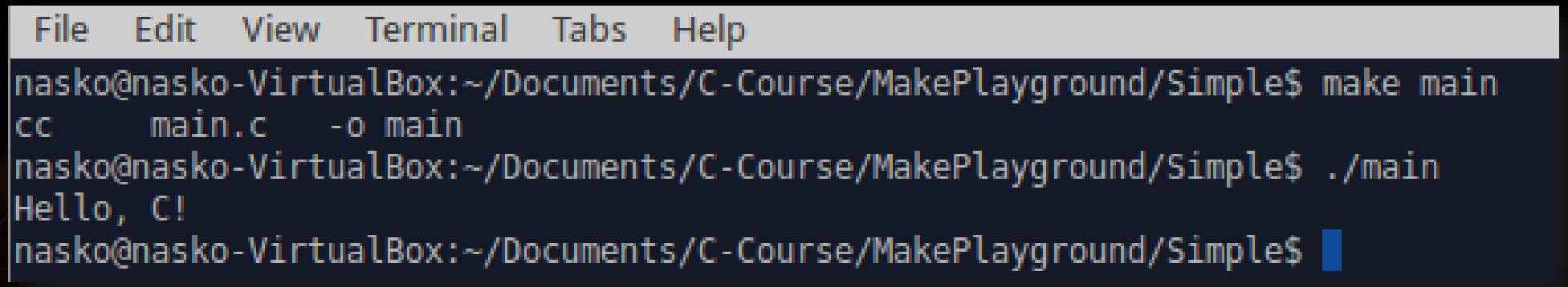
C Code Compilation and Execution

The Make Utility

- Under Linux C source code can be compiled with **make**:

```
make main
```

  - Produces executable called "main" with no extension
  - Can be run by executing `./{executable-name}`



C Compilation and Execution

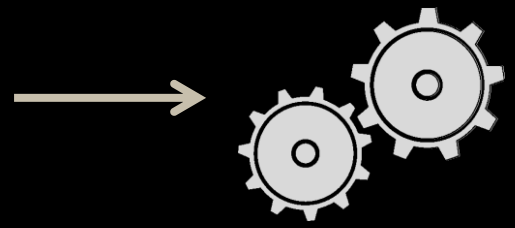
SOFTWARE UNIVERSITY FOUNDATION

main.c

```
#include <stdio.h>
int main()
{
    int a = 5;
    int b = 4;
    printf("%d", a + b);
    return 0;
}
```

(source code)

Compilation (gcc)

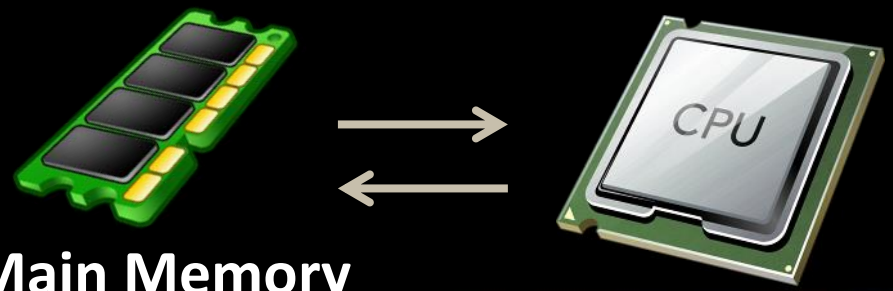


main

```
0x400570 mov DWORD PTR [rbp-0xc],0x5
0x400577 mov DWORD PTR [rbp-0x8],0x4
0x40057e mov eax,DWORD PTR [rbp-0x8]
0x400581 mov edx,DWORD PTR [rbp-0xc]
0x400584 add eax,edx
0x400586 mov DWORD PTR [rbp-0x4],eax
0x400589 mov eax,DWORD PTR [rbp-0x4]
0x400598 call 0x400410 <printf@plt>
```

(machine code)

Execution



Main Memory

Compiling and Running a C Program – Stages

1. Preprocessing – include-files, conditional compilation instructions and macros are processed by the preprocessor
2. Compilation – the compiler takes the source code and generates assembler (machine code)
3. Assembly – `.o` (object) files are generated
4. Linking – combines the `.o` files with external libraries to produce an executable file
5. Loading – the executable is loaded into main memory (RAM) and is granted a process to run on

Makefile

- A Makefile is a file that describes how to organize, compile and link several C source code files
  - E.g. we have the following two files:

main.c

```
#include <stdio.h>
#include "functions.c"
int main()
{
    int area = rect_area(4, 5);
    printf("%d\n", area);

    return 0;
}
```

functions.c

```
int rect_area(int a, int b)
{
    return a * b;
}
```

Calls function from other file

Makefile – Example

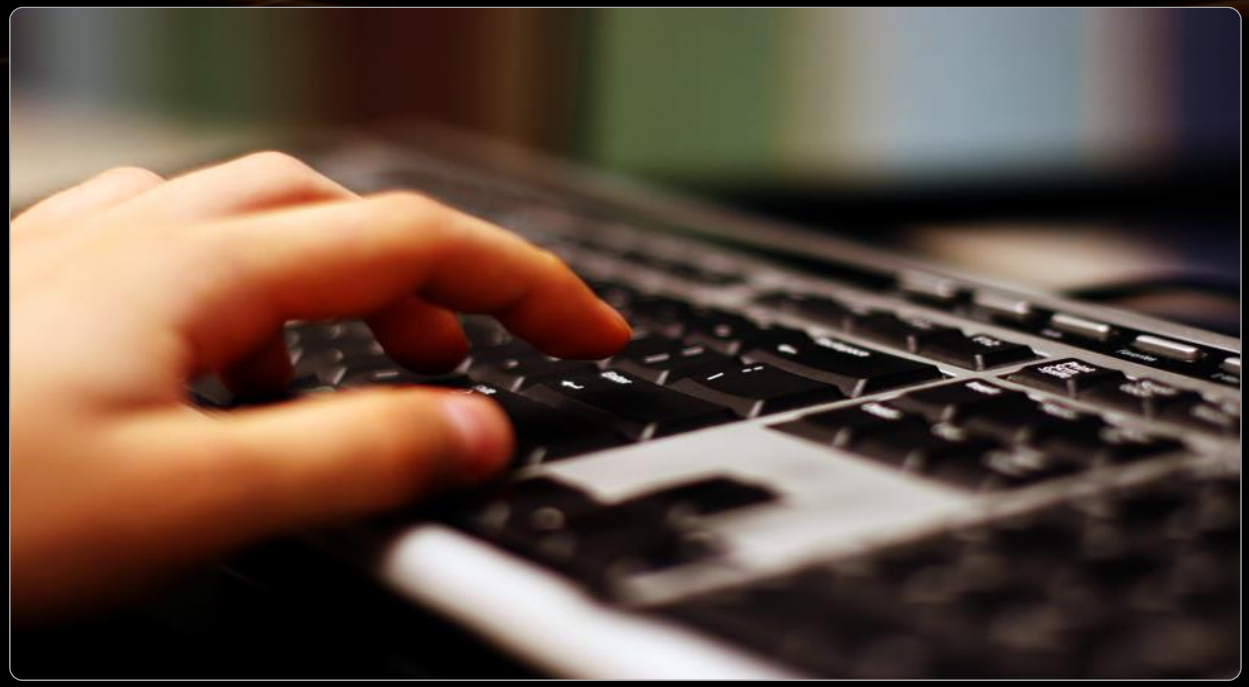
Tab (not space) indent is required

Makefile

```
program: main.o functions.o
	gcc main.o functions.o -o program
```

- Usually named Makefile
- Tells the compiler to:
  - Compile **`main.c`** and **`functions.c`** into object files
  - Link **`main.o`** and **`functions.o`** into a single executable – **`program`**
- Executed with the `make` command (no additional arguments)

SOFTWARE UNIVERSITY FOUNDATION



Compiling a C Program Manually

Live Demo

C Standards

- The C language has been standardized several times throughout its history
  - Official standards: ANSI C (1989), C99 (1999), C11 (2011)
  - Each new standard improves C and adds new functionality, e.g.:

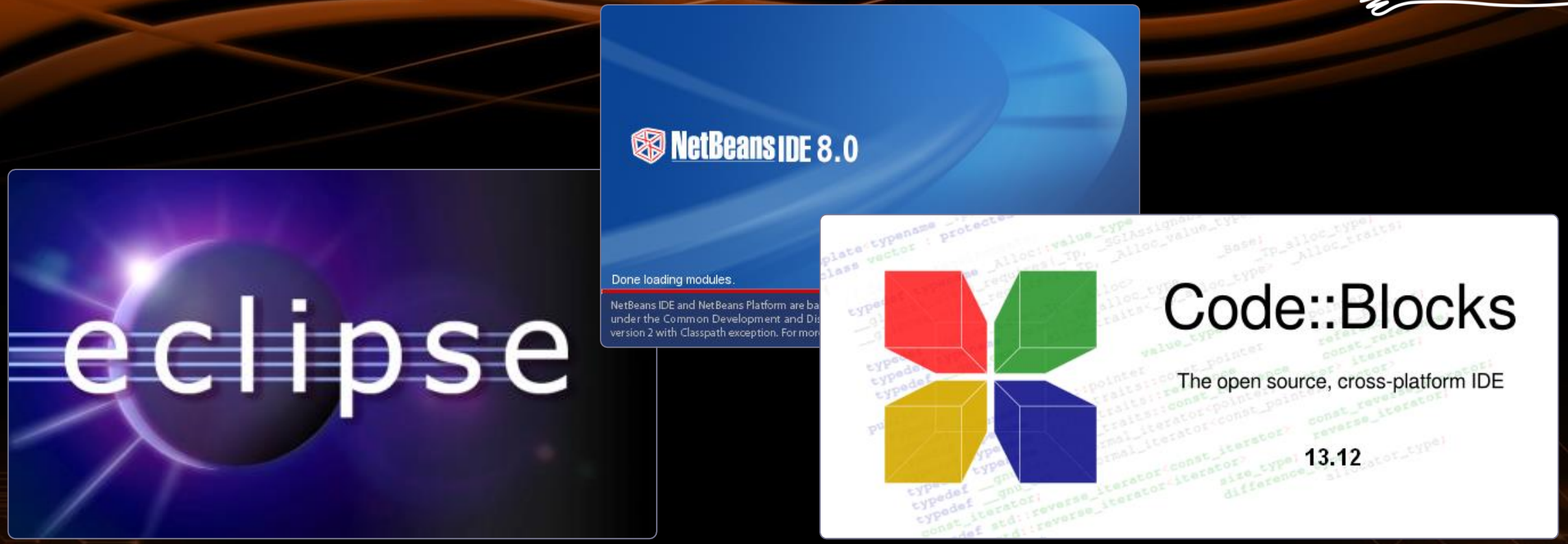
ANSI C for-loop declaration

```
int i;
for (i = 0; i < 10; i++)
```

C99 for-loop declaration

```
for (int i = 0; i < 10; i++)
```

  - Almost all modern compilers support C99 and C11
  - No reason to conform to ANSI C unless using MSVS compiler



C IDEs

Code::Blocks, Eclipse, Visual Studio

Eclipse CDT – IDE for C/C++ Developers

- Eclipse is popular IDE (Integrated Development Environment)
  - Written in Java
  - Supports Java, PHP, Python, JavaScript, C/C++, …
  - Features a code editor with inline documentation, syntax highlighting, auto-complete
  - Flexible debugger with lot's of options
  - Customizable UI with lot's of available plugins

https://eclipse.org/cdt/

Code::Blocks

- Code::Blocks is a free and open-source IDE oriented towards C/C++:
  - Supports a wide range of compilers (GCC, MinGW, Clang, Visual C++)
  - Features a code editor with syntax highlighting, code folding, debugger, etc.
  - Custom build configuration (not makefiles)
  - Has 3rd party plugins for pretty much anything

http://www.codeblocks.org/

NetBeans

- NetBeans is an open-source IDE
  - Supports Java, C++, PHP, HTML5
  - Developed by Oracle
  - Integrated features: code editor, debugger, profiler, GUI editor, source control tools, etc.
  - Supports inline documentation (Linux man pages)
  - Displays runtime errors on the console (unlike Eclipse CDT)

www.netbeans.org

SOFTWARE UNIVERSITY FOUNDATION

Creating, Compiling, Running and Debugging C Programs

NetBeans Live Demo

SOFTWARE UNIVERSITY FOUNDATION

C Documentation

C References

- CppReference
  - http://en.cppreference.com/w/
  - C/C++ reference with detailed explanations + code examples
- Linux man (manual) pages
  - Come with any Linux distribution
  - Displayed in the terminal via the command `man {function-name}`
  - Built into NetBeans and Eclipse code completion (opened with Ctrl + Space)

Introduction to C Programming

SOFTWARE UNIVERSITY FOUNDATION

Questions?

https://softuni.bg/trainings/1212/C-Programming-October-2015

License

- This course (slides, examples, demos, videos, homework, etc.) is licensed under the "Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International" license

- Attribution: this work may contain portions from
  - "Programming Basics" course by Software University under CC-BY-SA license

Free Trainings @ Software University

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