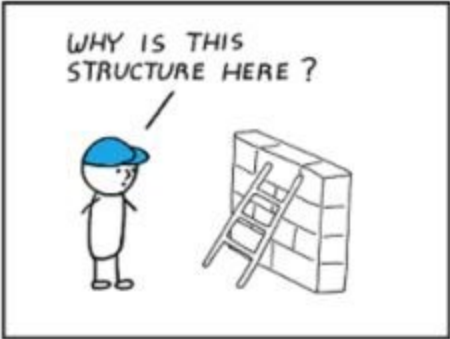
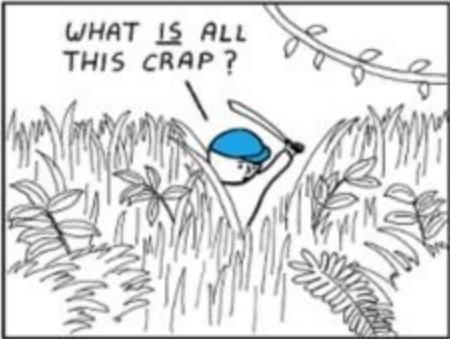


# **EMS30 Week 3 Les 2: Coding standards en static code analysis**

## Leerdoelen week 3 les 2. Je leert hoe je:

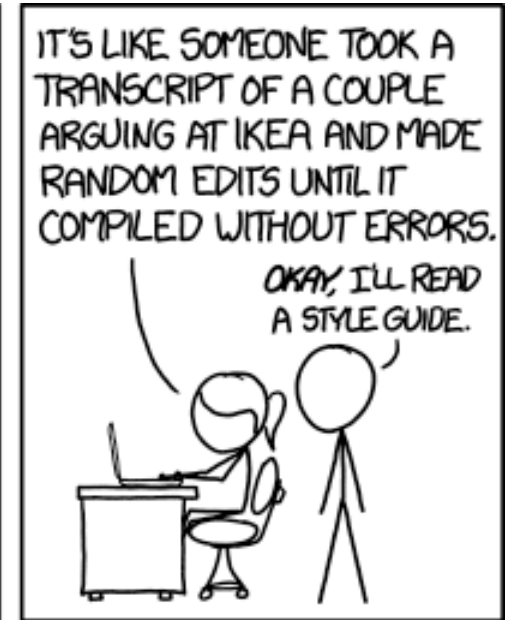
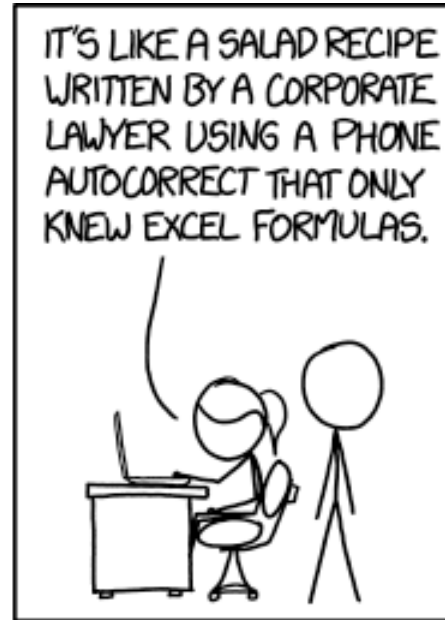
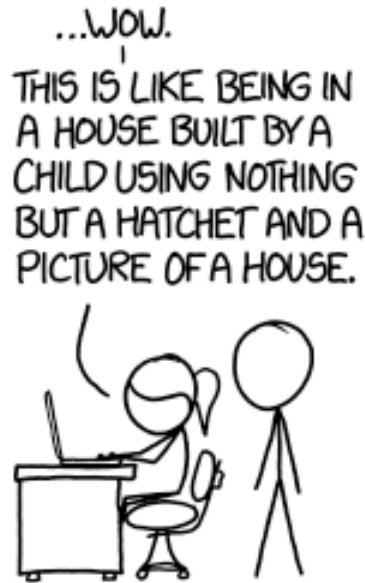
- gebruik kunt maken van verschillende **C coding standards**;
- gebruik kunt maken van verschillende **static code analysis tools**, waaronder **Cppcheck**, om de kwaliteit van je C-code te verbeteren.



I hate reading other people's code.

Bron: <https://zhuanlan.zhihu.com/p/20045208>

# Style guide



Bron: <https://xkcd.com/1513/>

Wat is het? Welke coding standards zijn beschikbaar voor C?

- MISRA (Motor Industry Software Reliability Association) MISRA C:2012:  
<https://www.misra.org.uk/> (betaald £15 pdf).
- Barr Group: Embedded C Coding Standard (2018):  
<https://barrgroup.com/Embedded-Systems/Books/Embedded-C-Coding-Standard> (gratis, zie [https://bitbucket.org/HR\\_ELEKTRO/ems30](https://bitbucket.org/HR_ELEKTRO/ems30))
- SEI (Software Engineering Institute) CERT (Computer Emergency Response Team) C Coding Standard (2016):  
<https://www.securecoding.cert.org/confluence/display/c/SEI+CERT+C+Coding+Standard> (gratis, zie [https://bitbucket.org/HR\\_ELEKTRO/ems30](https://bitbucket.org/HR_ELEKTRO/ems30))
- ... vele anderen ...

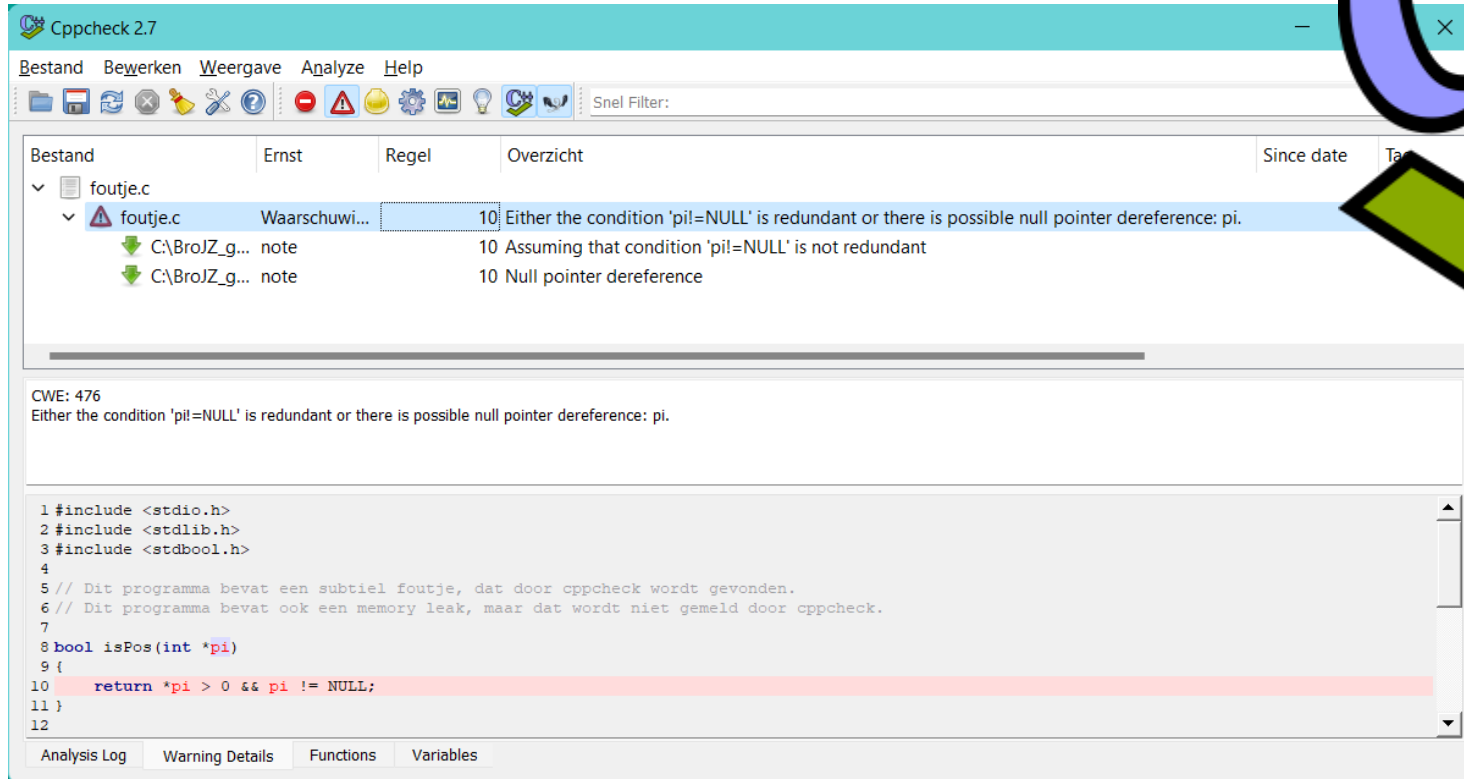
# Static code analysis

- Wat is het?
- Welke **gratis** tools zijn beschikbaar voor C?
  - **Cppcheck**
  - Lint, Splint, Adlint, OCLint, PC-Lint, ...
  - Clang –analyze
  - ...
- Welke **commerciële** tools zijn er voor C?
  - Coverty
  - Klocwork
  - Parasoft
  - Helix QAC
  - ...



# Cppcheck

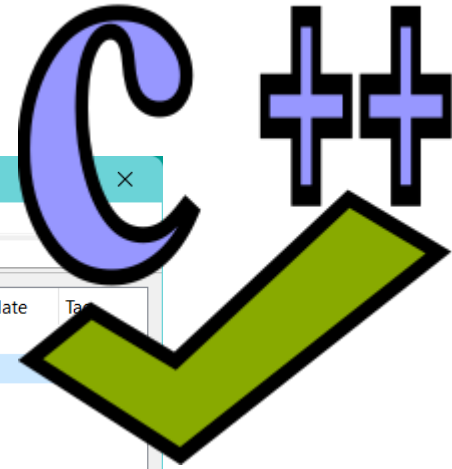
<http://cppcheck.sourceforge.net/>



The screenshot shows the Cppcheck 2.7 application window. The menu bar includes Bestand, Bewerken, Weergave, Analyze, and Help. The toolbar contains various icons for file operations and analysis. The main window displays a tree view of files, with 'foutje.c' selected. A warning is shown for 'foutje.c' with the message: 'Either the condition 'pi!=NULL' is redundant or there is possible null pointer dereference: pi.' The warning is categorized as 'Waarschuwi...' and has a severity of '10'. Below the warning, the CWE 476 is identified: 'Either the condition 'pi!=NULL' is redundant or there is possible null pointer dereference: pi.' The code snippet is shown in a text editor, with the line 'return \*pi > 0 && pi != NULL;' highlighted in red. The code includes standard headers and a function 'isPos' that takes a pointer 'pi' and returns a boolean value.

Bestand	Ernst	Regel	Overzicht	Since date	Ta
▼ foutje.c					
▼ ⚠ foutje.c	Waarschuwi...	10	Either the condition 'pi!=NULL' is redundant or there is possible null pointer dereference: pi.		
▼ C:\BroJZ_g...	note	10	Assuming that condition 'pi!=NULL' is not redundant		
▼ C:\BroJZ_g...	note	10	Null pointer dereference		

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <stdbool.h>
4
5 // Dit programma bevat een subtiel foutje, dat door cppcheck wordt gevonden.
6 // Dit programma bevat ook een memory leak, maar dat wordt niet gemeld door cppcheck.
7
8 bool isPos(int *pi)
9 {
10  return *pi > 0 && pi != NULL;
11 }
12
```



# Voorbeeld foutje.c

[https://bitbucket.org/HR\\_ELEKTRO/ems30/raw/master/Opdrachten/progs/cppcheck/foutje.c](https://bitbucket.org/HR_ELEKTRO/ems30/raw/master/Opdrachten/progs/cppcheck/foutje.c)

```
// Bepaal of pointer pi naar een positief getal wijst
bool isPos(int *pi)
{
    return *pi > 0 && pi != NULL;
}

int main(void)
{
    int *p = malloc(sizeof(int));
    if (p != NULL)
        *p = 15;
    if (isPos(p))
        printf("*p is positief, zoals verwacht!\n");
}
```



# Cppcheck in VS Code

**Cppcheck Plug-in**  
A plug-in for Cppcheck, capable of checking folders or editor tabs, shows output in the output channel, severity options available.  
NathanJ

**Gcov Viewer** 101ms  
Decorate C/C++ source files with code coverage information generated by gcov.  
Jacques Lucke

The screenshot shows the Explorer view in VS Code with a file tree. The file 'foutje.c' is selected, and a context menu is open over it. The menu items are:

- Open to the Side (Ctrl+Enter)
- Open With...
- Reveal in Explorer (Shift+Alt+R)
- Open in Integrated Terminal
- Select for Compare
- Open Timeline
- Cut (Ctrl+X)
- Copy (Ctrl+C)
- Download...
- Copy Path (Shift+Alt+C)
- Copy Relative Path (Ctrl+K Ctrl+Shift+C)
- Rename... (F2)
- Delete Permanently (Delete)
- Check C/C++ file** (highlighted)

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
/home/ems/Voorbeelden/cppCheck/foutje.c(17): error: Memory leak: p  
/home/ems/Voorbeelden/cppCheck/foutje.c(10): error: Null pointer dereference: pi
```

# CCS kan MISRA-C 2004 checken

Properties for line\_in\_2\_line\_out

type filter text

- Resource
  - General
- Build
  - ARM Compiler
    - Processor Options
    - Optimization
    - Include Options
    - ULP Advisor
    - Predefined Symbols
  - Advanced Options
    - Advanced Debug Option
    - Language Options
    - Parser Preprocessing Opt
    - Diagnostic Options
    - Runtime Model Options
    - Advanced Optimizations
    - Entry/Exit Hook Options
    - Feedback and Analysis O
    - Library Function Assump
    - Assembler Options
    - File Type Specifier
    - Directory Specifier
    - Default File Extensions
    - Command Files
    - MISRA-C:2004**
    - Supplemental Informatic
    - Miscellaneous

**MISRA-C:2004**

Configuration: Debug [ Active ] Manage Configurations...

Enable checking of MISRA-C:2004 rules (--check\_misra)

- 1: Environment
- 2: Language extensions
- 3: Documentation
- 4: Character sets
- 5: Identifiers
  - 5.2 [Required]: Identifiers in an inner scope shall not use the same name as an identifier in an outer scope.
  - 5.3 [Required]: A typedef name shall be a unique identifier.
  - 5.4 [Required]: A tag name shall be a unique identifier.
  - 5.6 [Advisory]: No identifier in one name space should have the same name as an identifier in another name space.
  - 5.7 [Advisory]: No identifier name should be reused
- 6: Types
- 7: Constants
- 8: Declarations and definitions

Command: "all"

Set severity of MISRA 'advisory' rule class (--misra\_advisory) warning

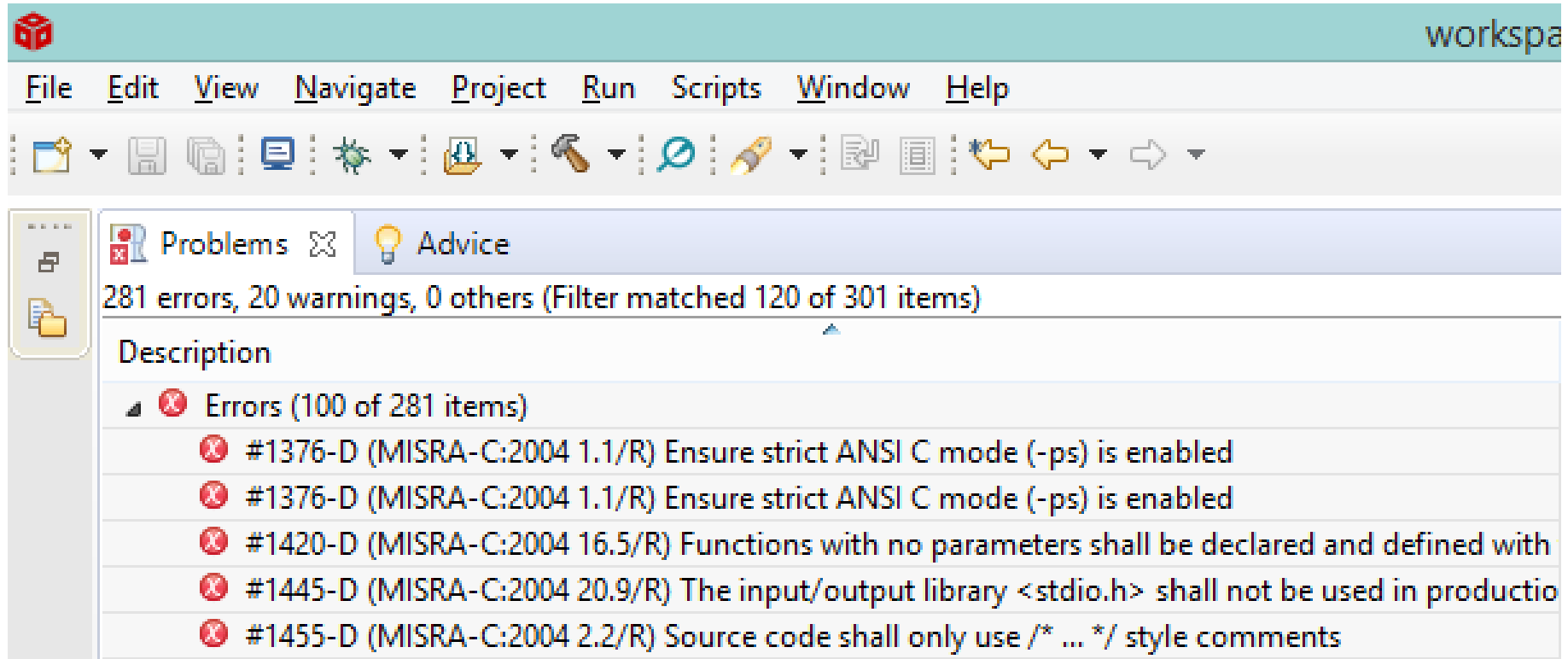
Set severity of MISRA 'required' rule class (--misra\_required) error

None All Required Advisory Expand All Collapse All

Apply and Close Cancel

# CCS MISRA-C 2004 code check

EMBEDDED SYSTEMS



The screenshot shows the CCS IDE workspace with the following elements:

- Menu Bar:** File, Edit, View, Navigate, Project, Run, Scripts, Window, Help
- Toolbar:** Includes icons for file operations (New, Save, Open), editing (Undo, Redo), and navigation (Home, End, etc.).
- Problems Window:**
  - Buttons: Problems (with error icon), Advice (with lightbulb icon)
  - Summary: 281 errors, 20 warnings, 0 others (Filter matched 120 of 301 items)
  - Section: Errors (100 of 281 items)
  - List of errors:
    - #1376-D (MISRA-C:2004 1.1/R) Ensure strict ANSI C mode (-ps) is enabled
    - #1376-D (MISRA-C:2004 1.1/R) Ensure strict ANSI C mode (-ps) is enabled
    - #1420-D (MISRA-C:2004 16.5/R) Functions with no parameters shall be declared and defined with
    - #1445-D (MISRA-C:2004 20.9/R) The input/output library <stdio.h> shall not be used in productio
    - #1455-D (MISRA-C:2004 2.2/R) Source code shall only use /\* ... \*/ style comments

# Volgende week...

## Werken aan [Eindopdracht 1](#)

```
> hallo
Onbekend commando: "hallo"
> nul
leds: groen = 0, geel = 0, rood = 0
> ze
Er zijn 2 opties: zes, zeven
> zes
leds: groen = 1, geel = 1, rood = 0
> schakelroodaan
leds: groen = 1, geel = 1, rood = 1
> schakelgroenuit
leds: groen = 0, geel = 1, rood = 1
> schakelgeelom
leds: groen = 0, geel = 0, rood = 1
> schakelgeelom
leds: groen = 0, geel = 1, rood = 1
> █
```

# Aan de slag!

Aan de slag met [Opdrachten Week 3 Les 2.pdf](#)

