

# TESINE

---

- **GOAL:** Write test cases for MISRA 2004 rules.
- MISRA 2004:
  - 141 rules
    - 121 compulsory
    - 129 automatically checkable
- What is a test case in this context?
  - A portion of code that **violates** a MISRA rule.
- 4 groups, 2 people for each group

# Example 1

---

- **Rule 2.2 (required):**

- Source code shall only use `/* ... */` style comments.

- **Test case :**

```
void misra_02_02 ( void )
```

```
{
```

```
    use_int32 ( 0 ); // Comment Not Compliant
```

```
}
```

# Example 2

---

- **Rule 9.3 (required):** In an enumerator list, the “=” construct shall not be used to explicitly initialise members other than the first, unless all items are explicitly initialised.
- **Test case:**

```
typedef enum
```

```
{
```

```
    mc2_0903_red_1    = 3,
```

```
    mc2_0903_blue_1,
```

```
    mc2_0903_green_1,
```

```
    mc2_0903_yellow_1 = 5    /* Not Compliant */
```

```
} mc2_0903_colour_1;
```

# Process

---

- Each group receives:
  - \_ A set of MISRA 2004 rules (some with explanation)
  - \_ URLs where to download the static analysis tools to use
  - \_ Detections table template
  - \_ Test cases template
- Each group provides the following deliverables:
  - \_ Test cases
  - \_ Static analysis output on your tests case
  - \_ Detections table

# Deliverables

---

- Tests cases: the test cases you wrote, respecting the standard format provided
- Output of static analysis on your tests cases
- Detections table: MISRA\_C 2004 violations detected by static analysis tools

All files (received/produced) in SVN repository

# Static analysis tool

---

- SPLINT 3.1.2 <http://splint.org/>
- CPPCHECK 1.40  
<http://sourceforge.net/projects/cppcheck/>
- Tools must be ran in standard mode

# Evaluation

---

- Evaluation criteria:
  - Tests cases correctness
  - Test cases completeness
  - Test cases redundancy
  - Compliance examples
  - Tests cases and summary table adherence to format