

Quality Programming

What is it all about?



Can you find the bug?

```
...
declare
  vertical_veloc_sensor: float;
  horizontal_veloc_sensor: float;
  vertical_veloc_bias: integer;
  horizontal_veloc_bias: integer;
...
begin
  declare
    pragma suppress(numeric_error, horizontal_veloc_bias);
  begin
    sensor_get(vertical_veloc_sensor);
    sensor_get(horizontal_veloc_sensor);
    vertical_veloc_bias := integer(vertical_veloc_sensor);
    horizontal_veloc_bias := integer(horizontal_veloc_sensor);
  ...
  exception
    when numeric_error => calculate_vertical_veloc();
    when others => use_irs10;
  end;
end irs2;
```



Ariane 5

- ▶ June 4th, 1996
- ▶ Flight 501
- ▶ Cost: ~1 Billion \$
- ▶ Software written in ADA



Ariane 5 – Inquiry

Israel^{tech}
challenge

- 37 seconds after liftoff – Ariane 5 reaches altitude of 3700 meter, and horizontal bias (velocity) of 32,768 (internal units).
- Operand error
- The OBC detected the error, and switched to SRI2.
- SRI2 had the same code, and therefore the same error.
- OBC read the output of SRI2 as flight commands, though it was error output.
- BOOM!

Ariane 5 – Inquiry

Israel^{tech}
challenge

- The flight control system is of a standard origin.
 - Inertial Reference System (SRI) – measures launcher's attitude, movement in space etc.
 - Data from SRI is transmitted to the OBC.
 - On-Board Computer (OBC) – executes flight program.
- Redundancy
 - Two SRIs working in parallel, with identical software and hardware.
 - One is active, the other is in stand-by.
 - Two OBCs.
- Ariane 5's software is almost identical to Ariane 4's.

Quality Programming

Israel^{tech}
challenge

- Working code is not good enough!
- We will write high-quality code
- We will learn new concepts while learning Python & C



Notes

Israel^{tech}
challenge

The error occurred in a part of the software that only performs alignment of the strap-down inertial platform. This software module computes meaningful results only before lift-off. As soon as the launcher lifts off, this function serves no purpose.

- Assuming that the code used in Ariane 4 will work.
 - Ariane 4 never reached speed of 32,768!
- Only 3 out of 7 variable overflows have been “protected”
 - Assuming nothing will happen for the others

Bugs

Israel^{tech}
challenge

- How can we avoid bugs?
 - We can't!**
- How can we deal with bugs?
 - Write tests
 - "Defensive coding"
 - Test return values for each function
 - Expect that the code will fail
 - Coding techniques
 - Code review
 - ...



Measures for quality?

Israel^{tech}
challenge

- Readability
- Modularity
- Robustness
- Testability
- Simplicity

```
if ((var=a( )) ==1) return var;  
  
if (should_fire_missile()) {  
    return TRUE;  
}
```

```
int getRandomNumber()  
{  
    return 4; // chosen by fair dice roll.  
              // guaranteed to be random.  
}
```

How can we avoid such cases?

Israel^{tech}
challenge

- Conventions to the rescue!
- Starting now, all your code should follow conventions

Hmm...

Israel^{tech}
challenge

```
if (missile.status = SHOULD_FIRE)  
{  
    missile.ignite = TRUE;  
}
```

An assignment expression has the value of the left operand after the assignment



Taste of conventions:
names

Israeltëch
challenge

- › **Always** meaningful and indicative
- › Variables and functions: lowercase, underscore delimited
- › **Right:**
 - `name_of_student = "John"`
- › **Wrong:**
 - `str_name_of_student = "John"`
 - `nameOfStudent = "John"`
 - `n = "John"`

PEP8

Israeltëch
challenge

- › Style guide for Python code
- › “code is read much more often than it is written”
- › You will receive a summary
- › PyCharm will help ☺

Taste of conventions:
names

Israeltëch
challenge

- › Constants: uppercase, underscore delimited
- › **Right:**
 - `BEST_COURSE = "ITC Fellows"`
- › **Wrong:**
 - `best_course = "ITC Fellows"`

Taste of conventions:
names

Israeltëch
challenge

- › **Right:**
 - `get_student_id(student_name)`
- › **Wrong:**
 - `getStudentId(studentName)`

Taste of conventions: documentation

Israel^{tech}
challenge

- Document using comments
- Document **why**, rather than **how** / **what**

‣ `x += 1` # Increment x

‣ `x += 1` # Add 1 for the NULL character



Taste of conventions: documentation

Israel^{tech}
challenge

- Document each **file** and each **function** using a documentation string

```
>>> def is_prime(number):
    """Returns True in case number is prime. Returns False otherwise."""
    for m in range(2, int(0.5*number)+1):
        if (number % m) == 0:
            return False
    return True

>>> help(is_prime)
Help on function is_prime in module __main__:

is_prime(number)
    Returns True in case number is prime. Returns False otherwise.
```

What's next?

Israel^{tech}
challenge

- Starting now – **follow conventions**
- Conventions summary
- PEP8
- Coming next – more topics

Taste of conventions: styling

Israel^{tech}
challenge

- Do not use “magic numbers”

‣ `print user_options[2]`

```
‣ # Command line arguments
INPUT_FILE = 1
USER_NAME = 2
...
print user_options[USER_NAME]
```



Good luck ☺

Israel^{tech}
challenge

