
ORACLE®

Testing and Testable code

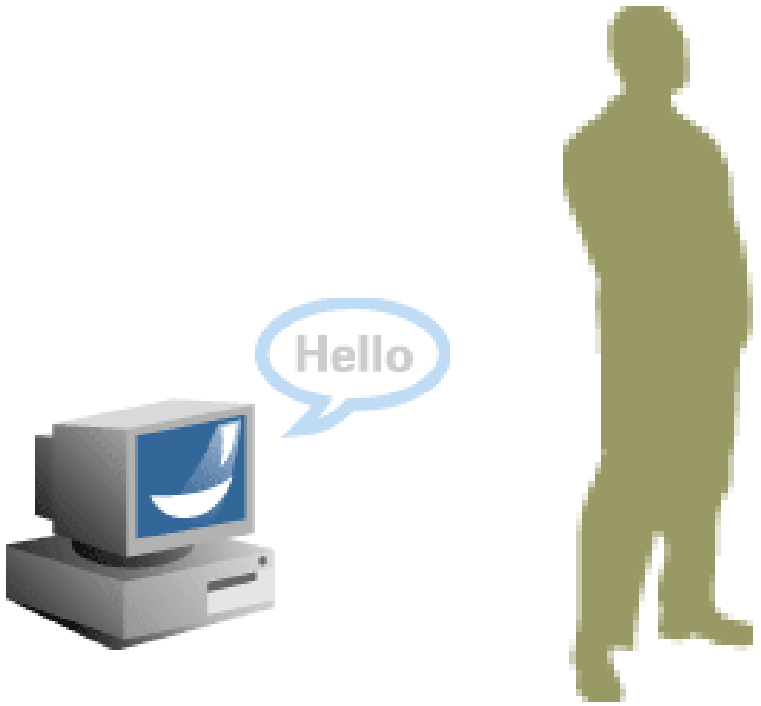
Jesper Wilhelmsson

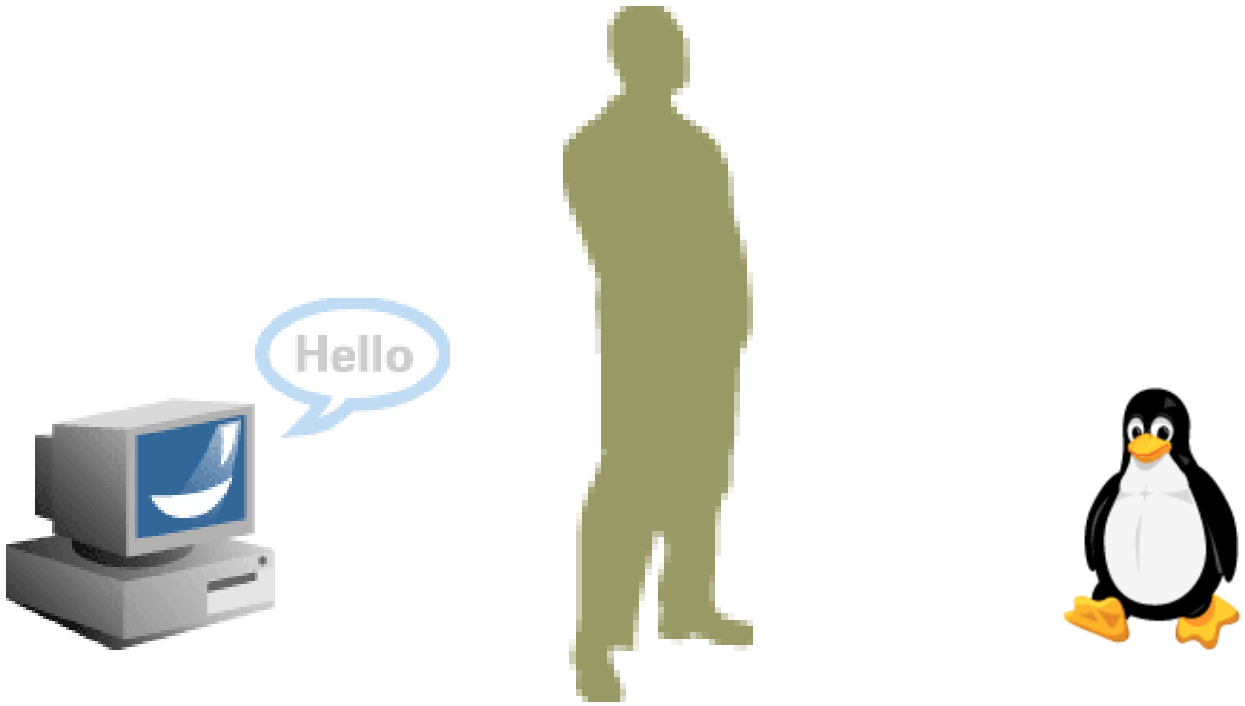
I'm ...

- Computer scientist from Pollax 1997 – 2002
  - Favorite courses: OS, PM2, OOP, KT2
- Ph.D student at IT-inst. 2002 – 2007
  - Automatic memory management
- Working at Oracle since 2007
  - More memory management...

Warranties


Hello


Hello


Hello

Mungosoft

Mungosoft

Mungosoft

M

Mungosoft


mEdit 3
New version
out now!

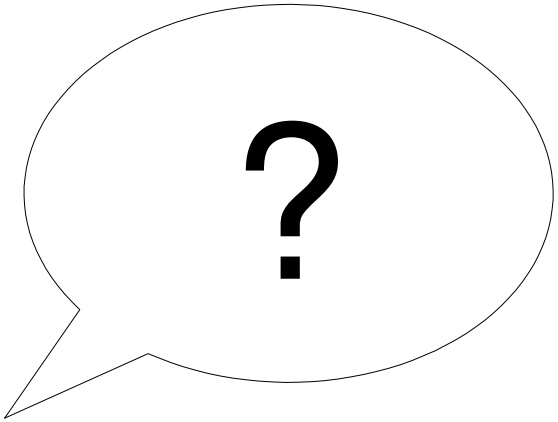
Verify that the module works

Test protocol

- Test 1:
  - Import image test01.png
  - Show the image
  - Rotate 90 degrees clockwise
  - Zoom 30%
  - Rotate 90 degrees counter clockwise
  - Show all images
  - Erase the image


Crash!


Crash!


?

```
typedef struct {
      char offset,kern, flags,flags2;
      int size;
      Bitvector* bv;
} bop;

bop* planes;

void bop_alloc(int s)
{
   int i,t=(64+s)*1024;
   planes=calloc(1024,64+s);
   for (i=0; i<t; i+=(s+2)) {
      ((int*)planes)[i]=0x008e;
      ((int*)planes)[i+1]=s;
   }
}
```

Verify that the code works

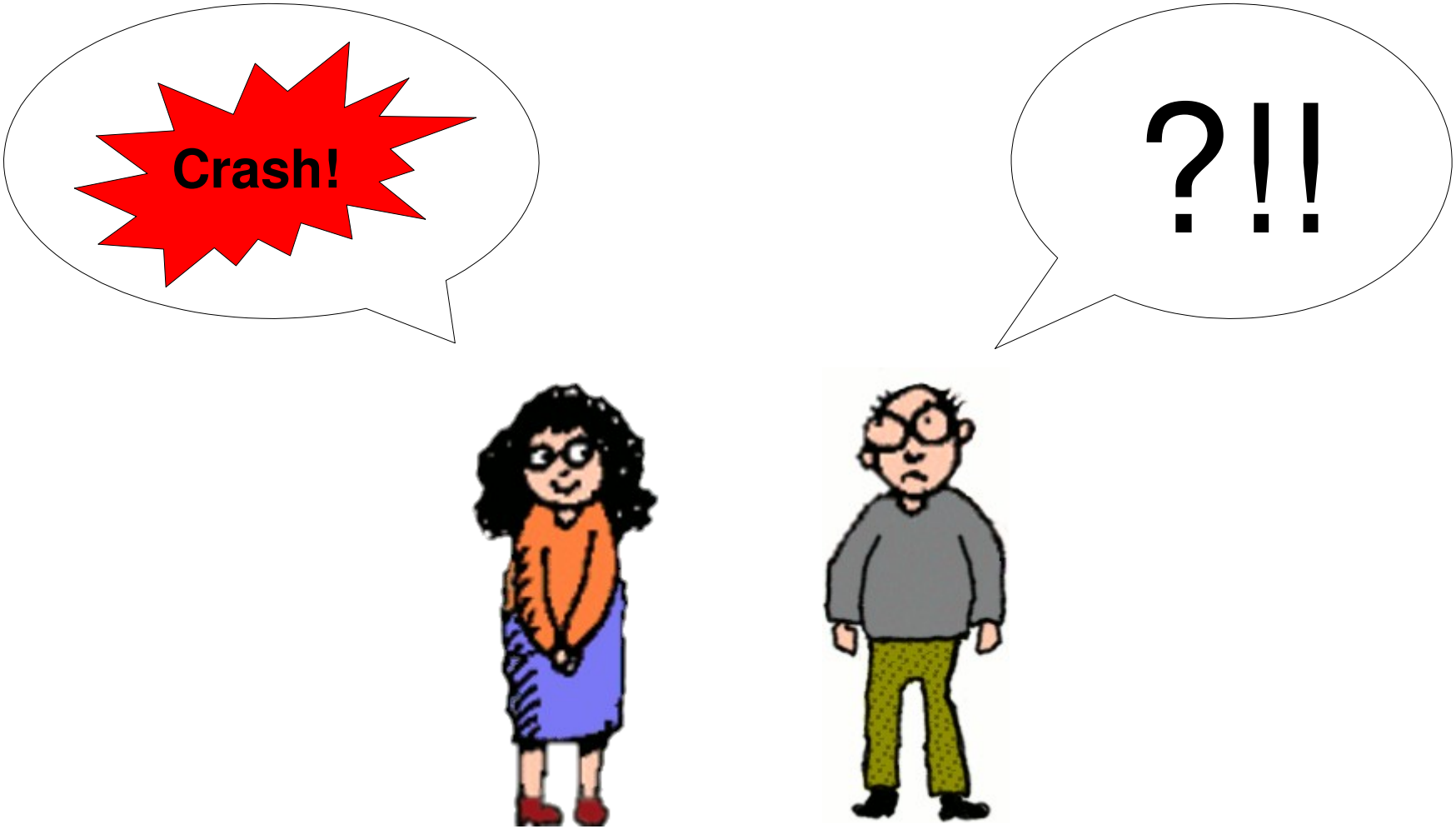
What code is most likely to contain bugs?

What code will you run if you follow the test protocol?

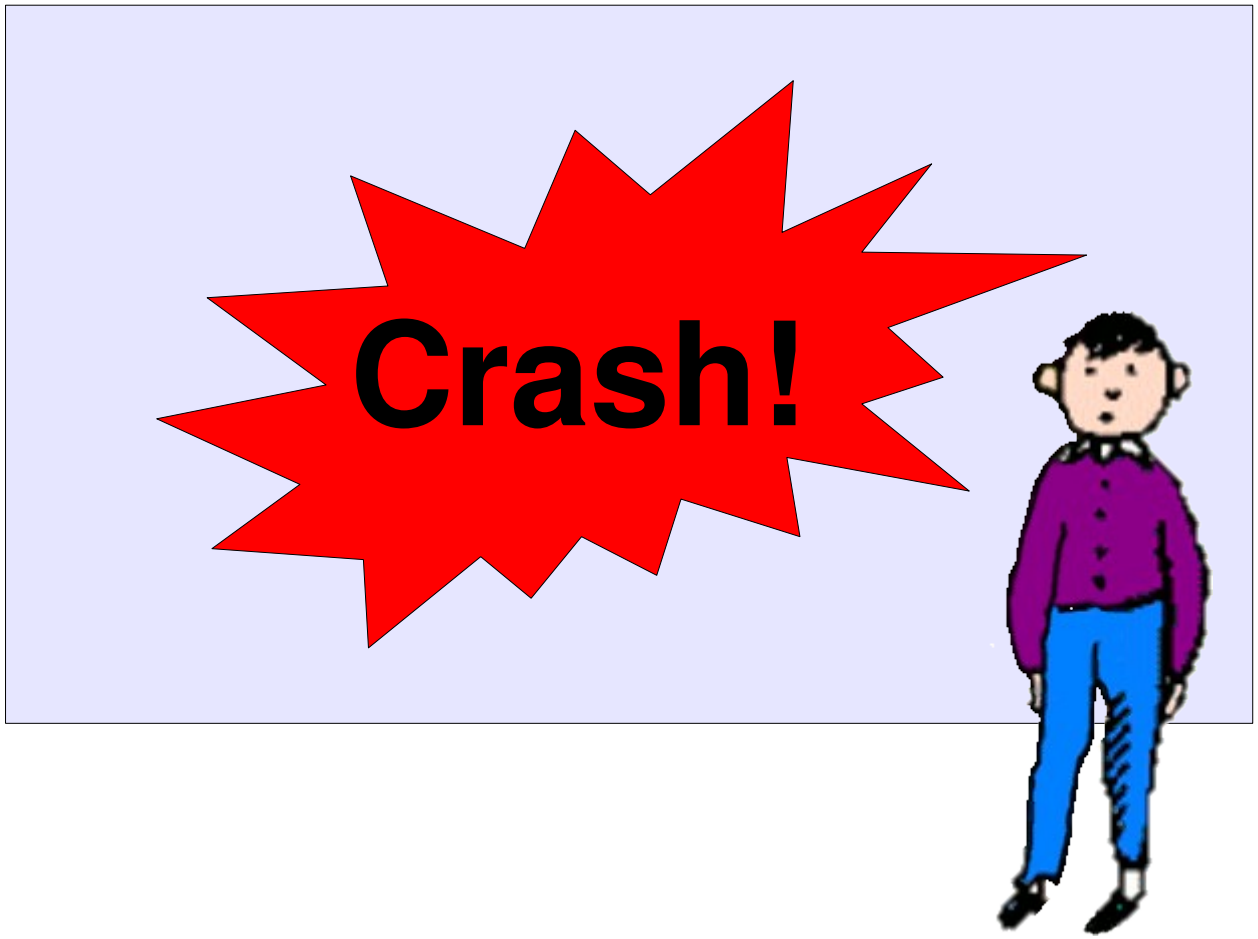
What code will the user run?

Verify that it doesn't work!

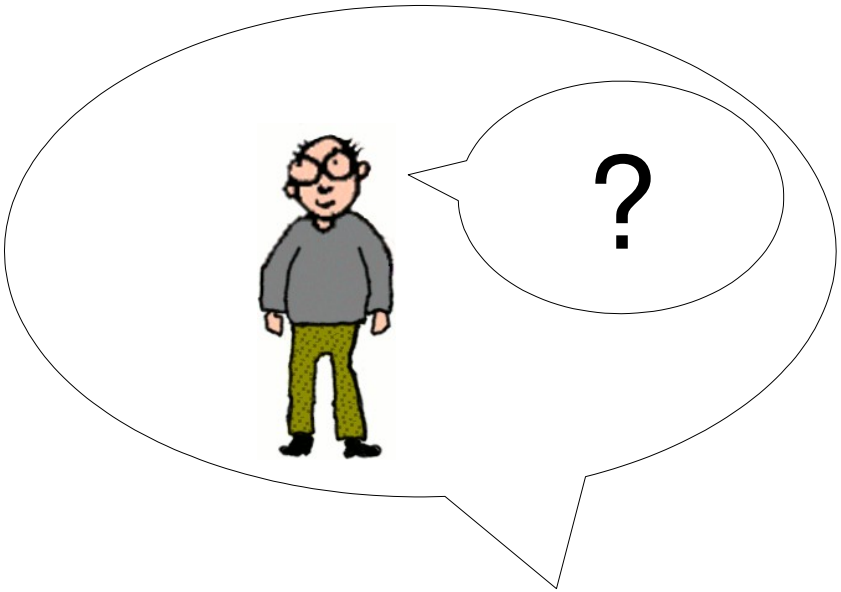
That's more like it!


Crash!
?!!


mEdit 3
New version
out now!


Crash!


Crash!
!



- Automate testing of old bugs. Each time a new bug is found, a new test should be created to reproduce it.
- Create a test framework that sends random instructions to the program.
- Create unit tests and integration tests for all modules.
- Run all tests nightly and generate a report containing the results.

Test types

- Unit testing
  - Test a small part of the program without any knowledge of the outside world
- Integration testing
  - Verify that different parts of the program works together

Assert

- Verify that a condition is fulfilled
  - Checked at runtime
  - May require special effort for performance

```
#include<assert.h>
assert(x == 5);
assert(getLength(myList) <
        MAX_LENGTH);
```

Who is responsible for testing?

Who is responsible for testing?

Everyone!

Who is responsible for testing?

Everyone!

... except the user ...

Never have more than five open bugs!

Readable code == Testable code?

- What should the source code look like?
  - Curly braces
  - CamelCase, lower_case, UPPER_CASE
  - Indenting
  - Variable declarations
  - Naming
  - Clear conditions
- Coding standard!
  - Everyone does it the same way

Testable code

- Modularize
  - Clean interfaces
  - No circular dependencies
- Avoid global states
  - Global variables
  - Global data
- Explicit dependencies
  - Make sure the arguments reflects the need of the function

Exercise

- Lots of programs depends on time and date. How should we test these?
  - Do we test at all?
  - Leap year?
  - Leap second?
  - Configurable?
  - Safe for the future?


Q&A


ORACLE®