**Example for Initialization Analysis**

class Test {

static void test(int p) {

int n;

p = p - 1;

if (p > 0) {

n = 100;

}

while (n != 0) {

[System](http://www.google.com/search?hl=en&q=allinurl%3ASystem+java.sun.com&bntI=I%27m%20Feeling%20Lucky).out.println(n);

n = n - p;

}

}

}

What does compiler do with the above program?

Corrected program:

We would like variables to be initialized on all execution paths. Otherwise, the program execution could be undesirable affected by the value that was in the variable initially:

* junk value that was there in the memory
* a value that an attacker purposely put into memory, especially in the presence of
  + [buffer overlow](http://en.wikipedia.org/wiki/buffer%20overlow)
  + [stack buffer overflow](http://en.wikipedia.org/wiki/stack%20buffer%20overflow)
  + [other cases of unusual language design decisions](http://www.php.net/manual/en/security.globals.php)
    - do not require variables to be declared before use
    - allow URL argument to set initial values of global variables
    - no checking if variable is initialized
    - a typo in variable name leads to attacker being able to set initial values of variables

A tool that checks PHP errors, including initialization: [Phantm](http://lara.epfl.ch/dokuwiki/phantm)