Potentially uninitialized local variable

Name: Potentially uninitialized local variable

Description: Reading from a local variable that has not been assigned to will typically yield garbage.

ID: cpp/uninitialized-local

Kind: problem

Severity: warning

Precision: medium

A local non-static variable of a non-class type has an undefined value before it is initialized. For example, it is incorrect to rely on an uninitialized integer to have the value `0`.

Recommendation

Review the code and consider whether the variable should have an initializer or whether some path through the program lacks an assignment to the variable.

Example

The function `absWrong` does not initialize the variable `j` in the case where `i = 0`. Functions `absCorrect1` and `absCorrect2` remedy this deficiency by adding an initializer and adding an assignment to one of the paths through the program, respectively.

```
1  int absWrong(int i) {
2      int j;
3      if (i > 0) {
4          j = i;
5      } else if (i < 0) {
6          j = -i;
7      }
8      return j; // wrong: j may not be initialized before use
9  }
10
11 int absCorrect1(int i) {
12     int j = 0;
13     if (i > 0) {
14         j = i;
15     } else if (i < 0) {
16         j = -i;
17     }
18     return j; // correct: j always initialized before use
19 }
20
21 int absCorrect2(int i) {
22     int j;
23     if (i > 0) {
24         j = i;
25     } else if (i < 0) {
26         j = -i;
27     } else {
28         j = 0;
29     }
30     return j; // correct: j always initialized before use
31 }
```

References

- ISO/IEC 9899:2011: Programming languages - C (Section 6.3.2.1).

- Common Weakness Enumeration: CWE-665.
- Common Weakness Enumeration: CWE-457.