The New C Standard (Excerpted material)

An Economic and Cultural Commentary

Derek M. Jones
derek@knosof.co.uk

3.17.2

indeterminate
value

indeterminate value

either an unspecified value or a trap representation

Commentary

object
initial value
indeterminate

This is the value objects have prior to being assigned one by an executing program. In practice it is a conceptual value because, in most implementations, an object's value representation makes use of all bit patterns available in its object representation (there are no spare bit patterns left to represent the indeterminate value).

unspeci-
fied value
trap repre-
sentation
reading is unde-
fined behavior

Accessing an object that has an unspecified value results in unspecified behavior. However, accessing an object having a trap representation can result in undefined behavior.

C++

Objects may have an indeterminate value. However, the standard does not explicitly say anything about the properties of this value.

4.1p1 *. . . , or if the object is uninitialized, a program that necessitates this conversion has undefined behavior.*

Common Implementations

A few execution time debugging environments tag storage that has not had a value stored into it so that read accesses to it cause a diagnostic to be issued.

Coding Guidelines

Many coding guideline documents contain wording to the effect that "indeterminate value shall not be used by a program." Developers do not intend to use such values and such usage is a fault. These coding guidelines are not intended to recommend against the use of constructs that are obviously faults.

guidelines
not faults

Example

```
1   extern int glob;
2
3   void f(void)
4   {
5   int int_loc;    /* Initial value indeterminate. */
6   unsigned char uc_loc;
7
8   /*
9    * The reasons behind the different status of the following
10   * two assignments is discussed elsewhere.
11   */
12  glob = int_loc; /* Indeterminate value, a trap representation. */
13  glob = uc_loc;  /* Indeterminate value, an unspecified value.  */
14  }
```

References