

N2687: Identifier - Primary expression

Javier A. Múgica - javier@aerotri.es

March 7th, 2021

Purpose

To clarify the explanation and to fix an error (as I see it) of when an identifier constitutes a primary expression.

Current text

In the section **6.5.1. Primary expressions**, it can be read:

Semantics

An identifier is a primary expression, provided it has been declared as designating an object (in which case it is an lvalue) or a function (in which case it is a function designator).¹³²⁾

A constant is a primary expression. Its type depends on its form and value, as detailed in 6.4.4.

¹³²⁾Thus, an undeclared identifier is a violation of the syntax.

It is supposed that explanations written as a footnote can be inferred from the main text (footnotes are not normative). However, the main text here implies in no way that an undeclared identifier is a violation of the syntax nor a violation of any kind, but merely that it does not constitute a primary expression. Indeed, any identifier which has not “been declared as designating an object [...] or a function” is on a par with respect to that explanation. So, just as it cannot be derived that an identifier designating a label, say, is a violation of the syntax, neither can that conclusion be drawn for an undeclared identifier. And it is recommended that footnotes be kept to a minimum.

On the other hand, that paragraph in its current recension delimits precisely when an identifier constitutes a primary expression: “An identifier is a primary expression, provided...”. It is written in the form “An object *so-and-so* is a primary expression, provided it satisfies *such-and-such*”. This wording, as I understand English, implies that if the identifier does not satisfy *such-and-such* it is not a primary expression. In particular, it implies that enumeration constants are not primary expressions.

But the paragraph which follows says that constants are primary expressions, and by looking at the section 6.4.4 referred to, and the point 6.4.4.3 therein, that paragraph implies that an enumeration constant is indeed a primary expression.

Proposed wording 1

Semantics

An identifier is a primary expression, provided it has been declared as designating **an enumeration constant (a case covered by the next paragraph)**, an object (in which case it is an lvalue) or a function (in which case it is a function designator).¹³²⁾ **An undeclared identifier is a violation of the syntax.**

A constant is a primary expression. Its type depends on its form and value, as detailed in 6.4.4.

~~¹³²⁾Thus, an undeclared identifier is a violation of the syntax.~~

Proposed wording 2

Semantics

An identifier **which is not an enumeration constant** is a primary expression, provided it has been declared as designating an object (in which case it is an lvalue) or a function (in which case it is a function designator).¹³²⁾ **An undeclared identifier is a violation of the syntax.**

A constant is a primary expression. Its type depends on its form and value, as detailed in 6.4.4.

~~¹³²⁾Thus, an undeclared identifier is a violation of the syntax.~~