

Updated Value Categories

C++ 17 introduces some updated value categories. This part helps familiarize with them. Read below to find out more!

WE'LL COVER THE FOLLOWING ^

- Let's look at the Diagram

In C++98/03, we had two basic categories of expressions:

- **lvalue** - an expression that can appear on the left-hand side of an assignment
- **rvalue** - an expression that can appear only on the right-hand side of an assignment

C++11 extended this taxonomy (due to the move semantics), with three more categories:

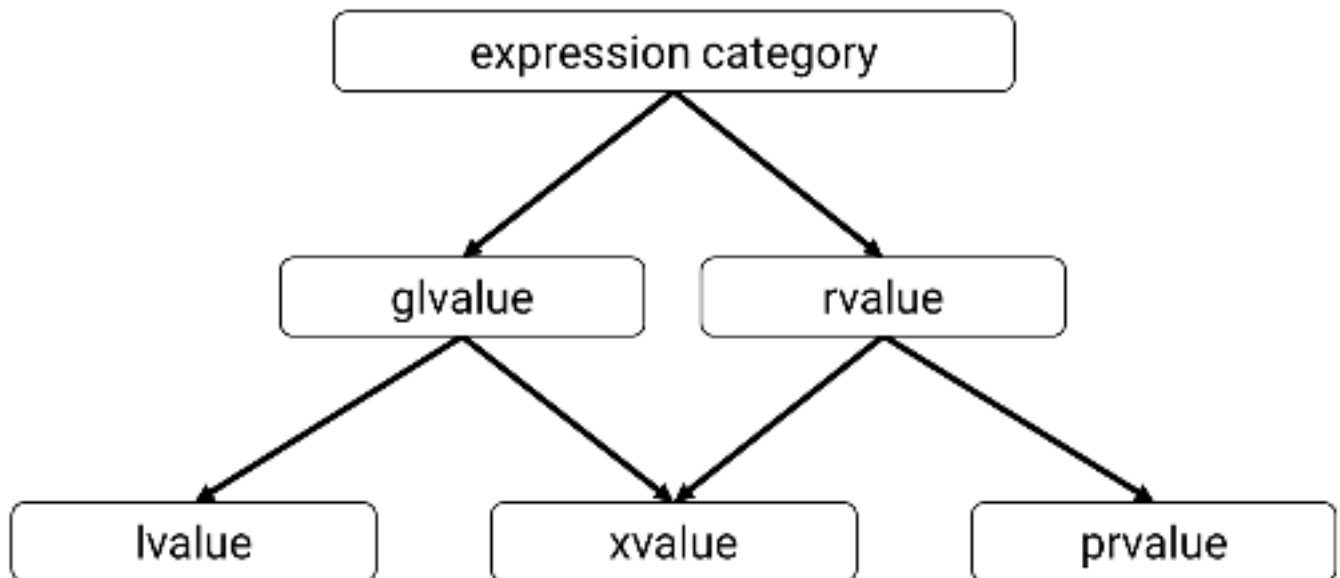
- **xvalue** - an eXpiring **lvalue**
- **prvalue** - a pure **rvalue**, an **xvalue**, a temporary object or subobject, or a value that is not associated with an object.
- **glvalue** - a generalised **lvalue**, which is an **lvalue** or an **xvalue**

Examples:

```
std::string str;  
str;           // lvalue  
42;           // prvalue  
str + "10"     // prvalue  
std::move(str); // xvalue
```

Let's look at the Diagram

The tree chart below gives a better overview of the categories:



There are three core categories (below with colloquial “definitions”):

- **lvalue** - an expression that has an identity, and which we can take the address of
- **xvalue** - "eXpiring **lvalue**" - an object that we can move from, which we can reuse. Usually, its lifetime ends soon
- **prvalue** - pure **rvalue** - something without a name, which we cannot take the address of, we can move from such expression

Now that you're familiar with **prvalue** and **lvalue**, in the next lesson, we will get into their further details.