*Exercise 6: L&R values (lvalue\_rvalue.cc)*

***Explained***

* **This exercise has us explore inheritance as it pertains to const virtual functions, copied objects, and transferred objects.** 
  + First and foremost, let’s remember that lvalues typically refer to data that is stored in memory (variables, references, and arrays).
  + Rvalues, on the other hand, are temporary things that are destroyed without having to be stored (Ex: 5 or int{5} are both rvalues).
  + Also, “&x” in a function header refers to a parameter being passed be reference (typically an Lvalue).
  + “&&x” in a function header refers to an expected **RVALUE**.
* **This output we see here is as follows:**
  + func(int && x) rvalue ref | **5 is merely an Rvalue**
  + func(int && x) rvalue ref | **2 + 3 is never stored so it too is an Rvalue**
  + func(int && x) rvalue ref | **int{5} is never stored so it too is an Rvalue**
  + func(int & x) lvalue ref | **our first Lvalue! Why? Because we explicitly created the variable “int x = 5”**

Text

Description automatically generated