*Exercise 5: Slicing (slicing.cc)*

***Explained***

* **This exercise has us explore inheritance as it pertains to const virtual functions, copied objects, and transferred objects.**
* **This exercise has us explore inheritance as it pertains to const virtual functions, copied objects, and transferred objects.** 
  + We first create a Cow object c.
  + Next an Animal is created using the copy constructor to copy our Cow object c (meaning that both c and this new animal are pointing to the same address).
  + Now we create another animal object, but this one should have its own address, but all its content will also be the same as Cow object c.
  + Finally “poke()” is called with all three of our different objects respectively passed as arguments (**Wow, now when we think it through it makes complete logical sense):**
    - Poke(c) = “moo” | ***Because according to our virtual function only Cows can print “moo”***
    - Poke(cr) = “moo” | ***Since we are creating this object BY REFERENCE it makes perfect sense that it should behave and be treated exactly like a cow since it is an exact copy of a cow that points to the same address.***
    - Poke(cs) = “I don’t know how to make a sound.” | Since this Animal object is a literal copy that even has its own address, even though it’s content is the same, it is outwardly treated like a regular old animal object as expected!

**Text

Description automatically generated**