CSI 32 LECTURE NOTES (Ojakian)

Topic 9: References, Addresses, Pointers

OUTLINE
PRIMER: 2.4, 2.5, 4.2, 5.11
TRANSITION GUIDE: 8.1, 8.3, 8.4

1. Python References
2. C++ Pointers

---

1. Addresses

   The byte location.

   (a) In C++ use the "and" symbol to get the address in HEX.
   (b) In Python use "id" to get address in Decimal.
   (c) Compare addresses in C++ versus Python: static versus dynamic.
   (d) Note space between integers.
   (e) Recall Hex.

2. References

   References have no soul! Just following someone else ...

   (a) Recall Python: Most everything by reference (exceptions: int, float, str)
   (b) Reference Variables in C++: Use same symbol as Address, but used in declaration!
   (c) Try uninitialized reference.

3. C++ Pointers

   Reference variables monogomous; pointers not!

   (a) Declare with * *after type.*
   (b) Pointer variable has data which is a memory address for the given type
   (c) Stores memory address of variable.
   (d) Access data at pointer by * in front - called *dereferencing*: Note the two **different** uses of star operator:
      i. In declaration
      ii. In de-referencing

4. Constant Modifier - `const`

   (a) Declaring a variable with a fixed value (must be initialized and can not be changed).

   PROBLEM 1. *Declare a* `const` *variable, then try to change it. Also what happens if not initialized?*