Please turn in answers to the following questions next Monday, in class.

1. Rewrite the following expressions in a simpler or more elegant way.
   Assume i is a non-negative int and result and b are type bool.
   
   (a)  
   \[ i = i - 19 \]

   (b)  
   ```python
   if True:
       print("Hello, world!")
   ```

   (c)  
   ```python
   a = (b == False)
   ```

   (d)  
   ```python
   if result:
       return True
   else:
       return False
   ```

   (e)  
   ```python
   # assume i is non-negative
   while i >= 100:
       i = i - 100
   ```

2. The relationships between values.

   (a) Explain the difference between \( o1 \text{ is } o2 \) and \( o1 \text{ == } o2 \).

   (b) Possible or not: \( (o1 \text{ is } o2) \text{ and not } (o1 \text{ == } o2) \)? Explain.

   (c) Write Python that sets two non-empty lists \( l1 \) and \( l2 \) in such a way that
   ```python
   (l1 == l2) \text{ and not } (l1 \text{ is } l2)
   ```
3. Consider these concepts related to abstraction.

(a) We use functions or procedures to support procedural abstraction. What part of a function is its public interface? What part is its private implementation?

(b) This week we're learning about modules. A module is a file that contains a collection of functions and values that support a single purpose (e.g., the math module). Modules typically have an \_\_all\_ variable that controls how symbols are imported. Explain how a module is an abstraction.