From Section 6.8 in *Python Programming: An Introduction to Computer Science*:

- **True/False: 1-10**
  1) False - programmer define their own functions all the time
  2) False - you can call a function many times throughout a program. For example, we call the print function many times at many different places in our assignments
  3) True
  4) True - if we consider None a value
  5) False - in Python, all parameters are passed by value, but if a value is a mutable object, you can make changes to that object that will remain after the function call
  6) False - when multiple values are returned from a function, they are assigned by position
  7) If the parameter is mutable, for example a list, the contents of the list can be modified
  8) True
  9) the memory occupied by local function variables is reclaimed when the function finishes.
  10) False - the program should be readable, so if making the program longer makes it more understandable, that is ok.

- **Multiple Choice: 2, 3, 6-10**
  2) def
  3) return
  6) by value - params are passed to functions by value
  7) d, it’s not about the ego
  8) a function returning a value should be called from an expression; because the equal symbol is being used
  9) None - try it yourself
  10) mutable

- **Discussion: 1, 3-5**
  1) two motivations for defining functions are: reduce code duplication and to make a program more self documenting
  3) a) the purpose of parameters is to allow explicit input into your function
     b) A formal parameter is a list of variable names. Actual parameters are the arguments passed through when the function is called. Actual parameters are matched up with formal parameters by position
     c) ordinary variables are local to the function or can be discarded or overwritten after the function ends, while parameters tend to have a larger scope

- **Programming Exercises: 1, 2, 4, 7, 10**
  See CMPS5P assignment 6 code Python file under attachments
  (to be posted shortly)

From Section 8 in *Python Programming: An Introduction to Computer Science*:

- **True/False: 7-10**
  7) False (only one operand has to be true),
  8) True,
  9) False (not (a or b) equates to not a and not b),
  10) True

- **Multiple Choice: 6, 8, 9**
  6) c (not (a and b) == not(a) or not(b)),

---

1 of 3
8) b (TFT should be TFF),
9)d (FTF should be FTT)

- Discussion: 2
  See pictures below

<table>
<thead>
<tr>
<th>a)</th>
<th>P</th>
<th>Q</th>
<th>P and Q</th>
<th>not (P and Q)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td></td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td></td>
<td>F</td>
<td>T</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td></td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>T</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>b)</th>
<th>P</th>
<th>Q</th>
<th>not P</th>
<th>(not P) and Q</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td></td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td></td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td></td>
<td>F</td>
<td>F</td>
<td>T</td>
<td>R</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>c)</th>
<th>P</th>
<th>Q</th>
<th>not P</th>
<th>not Q</th>
<th>not P or not Q</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td></td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td></td>
<td>F</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td></td>
<td>F</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
</tbody>
</table>

- Programming Exercises: 4-8
  See CMPS5P assignment 5 code Python file under attachments

From Section 11 in *Python Programming: An Introduction to Computer Science:*

- True/False:
  10) False (a Python dictionary is not a type of sequence)
• Multiple Choice: 9, 10
  9) c (sort),
  10) dictionary.items() returns (sequence of tuples)

• Discussion: 2
  Use shell to get output.