**Assignment 2 - Solutions**

1. **What are the two values of the Boolean data type? How do you write them?**
   1. The two Boolean values are: `True` and `False` (capitalized).
2. **What are the three different types of Boolean operators?**
   1. `and`
   2. `or`
   3. `not`
3. **Boolean operator truth tables:**

\* `and`:

| A | B | A and B |

| ----- | ----- | ------- |

| True | True | True |

| True | False | False |

| False | True | False |

| False | False | False |

\* `or`:

| A | B | A or B |

| ----- | ----- | ------ |

| True | True | True |

| True | False | True |

| False | True | True |

| False | False | False |

\* `not`:

| A | not A |

| ----- | ----- |

| True | False |

| False | True |

1. **Values of the expressions:**
   1. `(5 > 4) and (3 == 5)` → `False`
   2. `not (5 > 4)` → `False`
   3. `(5 > 4) or (3 == 5)` → `True`
   4. `not ((5 > 4) or (3 == 5))` → `False`
   5. `(True and True) and (True == False)` → `False`
   6. `(not False) or (not True)` → `True`
2. **Six comparison operators:**
   1. `==` (equal to)
   2. `!=` (not equal to)
   3. `<` (less than)
   4. `>` (greater than)
   5. `<=` (less than or equal to)
   6. `>=` (greater than or equal to)
3. **Difference between equal to (`==`) and assignment (`=`):**
   1. `=` is used to assign a value to a variable: `x = 5`
   2. `==` is used to compare two values: `x == 5`

**7. Identify the three blocks in the code:**

spam = 0

if spam == 10:

print('eggs') # Block 1

if spam > 5:

print('bacon') # Block 2

else:

print('ham') # Block 3

print('spam')

print('spam')

**8. Code using if-elif-else:**

spam = 1

if spam == 1:

print("Hello")

elif spam == 2:

print("Howdy")

else:

print("Greetings!")

1. **If your program is stuck in an endless loop, press:**
   1. `Ctrl + C` (in most command-line interfaces)
2. **Difference between `break` and `continue`:**
   1. `break` exits the loop entirely.
   2. `continue` skips the rest of the loop and starts the next iteration.
3. **Difference between `range(10)`, `range(0, 10)`, and `range(0, 10, 1)`:**
   1. All are equivalent. They generate numbers from 0 to 9.
   2. Syntax difference:
      1. `range(10)` → starts at 0, steps by 1
      2. `range(0, 10)` → explicitly defines start
      3. `range(0, 10, 1)` → explicitly defines start and step

**12. Programs printing 1 to 10:**

\* Using `for` loop:

for i in range(1, 11):

print(i)

\* Using `while` loop:

i = 1

while i <= 10:

print(i)

i += 1

**13. Calling function `bacon()` from module `spam`:**

import spam

spam.bacon()