

Assignment 2 - Solutions

1. What are the two values of the Boolean data type? How do you write them?

- a) The two Boolean values are: `True` and `False` (capitalized).

2. What are the three different types of Boolean operators?

- a) `and`
b) `or`
c) `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

4. Values of the expressions:

- a) `(5 > 4) and (3 == 5)` → `False`
b) `not (5 > 4)` → `False`
c) `(5 > 4) or (3 == 5)` → `True`
d) `not ((5 > 4) or (3 == 5))` → `False`
e) `(True and True) and (True == False)` → `False`
f) `(not False) or (not True)` → `True`

5. Six comparison operators:

- a) `==` (equal to)
b) `!=` (not equal to)
c) `<` (less than)
d) `>` (greater than)
e) `<=` (less than or equal to)
f) `>=` (greater than or equal to)

6. Difference between equal to (`==`) and assignment (`=`):

- a) `=` is used to assign a value to a variable: `x = 5`
b) `==` 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!")
```

9. If your program is stuck in an endless loop, press:

- a) `Ctrl + C` (in most command-line interfaces)

10. Difference between `break` and `continue`:

- a) `break` exits the loop entirely.
- b) `continue` skips the rest of the loop and starts the next iteration.

11. Difference between `range(10)`, `range(0, 10)`, and `range(0, 10, 1)`:

- a) All are equivalent. They generate numbers from 0 to 9.
- b) Syntax difference:
 - i. `range(10)` → starts at 0, steps by 1
 - ii. `range(0, 10)` → explicitly defines start
 - iii. `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()
```