ASSIGNMENT 2

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

solution

Two values of Boolean data type is True and False. First letter of Boolean data type should be capital letter other are lower case

True: affirmative condition

False: negative condition

Example

x=10

If x>5: (this statement gives True)

If x >25: (this statement give False)

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

Solution

Boolean operators are also known as logical operators. They are used commonly used in conditional statements, logical expressions

and =return true only when both operant are true

or = return true only when any one operant true

not = it is an unary operator which will negate the operant

3. Make a list of each Boolean operator's truth tables (i.e. every possible combination of Boolean values for the operator and what it evaluates).

Solution

A	not A
True	False
False	True

List of Boolean operators
or
not
and

and		
A	B	A and B
True	True	True
False	True	False
True	False	False
False	False	False

or		
A	B	A OR B
True	True	True
False	True	True
True	False	True
False	False	False

4. What are the values of the following expressions?

(5 > 4) and (3 == 5)

not (5 > 4)

(5 > 4) or (3 == 5)

not ((5 > 4) or (3 == 5))

(True and True) and (True == False)

(not False) or (not True)

solution

(5 > 4) and (3 == 5)	# False
not (5 > 4)	# False
(5 > 4) or (3 == 5)	# True
not ((5 > 4) or (3 == 5))	# False
(True and True) and (True == False)	# False
(not False) or (not True)	# True

5. What are the six comparison operators?

solution

1. Equal (==): Checks if two values are equal.

Example: 5 == 5 evaluates to True

2.Not Equal (!=): Checks if two values are not equal.

Example: 5 != 3 evaluates to True.

3. Greater Than (>): Checks if the left value is greater than the right value.

Example: 5 > 3 evaluates to True

4. Less Than (<): Checks if the left value is less than the right value.

 Example: 3 < 5evaluates to True

5. Greater Than or Equal To (>=): Checks if the left value is greater than or equal to the right value

Example: 5 >= 5evaluates to True.

6.Less Than or Equal To (<=): Checks if the left value is less than or equal to the right value.
Example: `3 <= 5` evaluates to `True`.

6. How do you tell the difference between the equal to and assignment operators? Describe a condition and when you would use one.
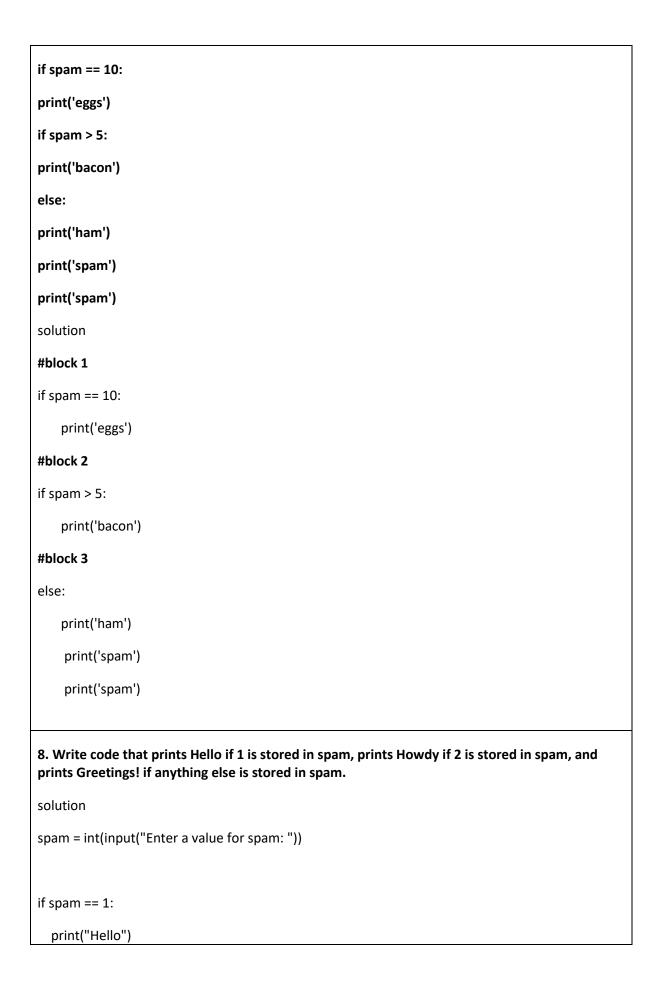
Solution

Assignment operator is used to assigning value to variable

Equal to used (==).It is a comparison operator that returns True if the values on both sides are equal and False otherwise

7. Identify the three blocks in this code:

spam = 0

```
if spam == 10:

print('eggs')

if spam > 5:

print('bacon')

else:

print('ham')

print('spam')

print('spam')
```

solution

#block 1

```
if spam == 10:
    print('eggs')
```

#block 2

```
if spam > 5:
    print('bacon')
```

#block 3

```
else:
    print('ham')
     print('spam')
     print('spam')
```

8. Write code that prints Hello if 1 is stored in spam, prints Howdy if 2 is stored in spam, and prints Greetings! if anything else is stored in spam.

solution

```
spam = int(input("Enter a value for spam: "))


if spam == 1:
    print("Hello")
```

```
elif spam == 2:

    print("Howdy")

else:

    print("Greetings!")
```

9.If your programme is stuck in an endless loop, what keys you'll press?

Solution: effectiveness of these key combinations may vary depending on specific environment and the nature of the program. In some cases, we need to forcefully terminate the program using operating system's task manager or process manager if the program doesn't respond to the standard interrupt signals

Some of the key are

Ctrl + C (Control key and C key together): On most systems, pressing Ctrl + C in the terminal or command prompt will send an interrupt signal to the running program, causing it to terminate. This is a common way to break out of an infinite loop.

Ctrl + Break: On some systems or IDEs, we might need to press Ctrl + Break instead of Ctrl + C to interrupt the program.

Ctrl + Z (Windows):In Windows command prompt, pressing Ctrl + Z can sometimes stop or pause a running program

10. How can you tell the difference between break and continue?

Solution

In Python, break and continue are two control flow statements used within loops to alter the program's flow. difference between break and continue is that break is used to exit the loop entirely, while continue is used to skip the current iteration and continue with the next iteration of the loop. Both statements are valuable tools for controlling the flow of loops and achieving specific behaviours within your code.

break Statement: The break statement is used to exit the current loop prematurely, regardless of whether the loop condition has been met or not. When break is encountered, the loop terminates immediately, and the program continues executing the code after the loop.It is typically used to exit a loop early when a certain condition is met or when a specific goal has been achieved

Example:

```
for i in range(1, 6):

    if i == 3:

        break                    # This will exit the loop when i is 3
```

```
    print(i)
```

continue Statement: The continue statement is used to skip the current iteration of a loop and move to the next iteration. When continue is encountered, the remaining code within the current iteration is skipped, and the loop proceeds to the next iteration.It is typically used when you want to skip certain iterations based on a condition but continue with the loop.

```
  for i in range(1, 6):

    if i == 3:

        continue  # This will skip the iteration when i is 3

    print(i)
```

11. In a for loop, what is the difference between range(10), range(0, 10), and range(0, 10, 1)?

solution

 all three of these range expressions will result in the same sequence of numbers 0, 1, 2, 3, 4, 5, 6, 7, 8, 9. we can use any of them interchangeably in a for loop to iterate over the same range of values. The default start value is 0, and the default step value is 1 when they are not explicitly specified.

```
for i in range(10):

    print(i)
```

```
for i in range(0,10):

    print(i)
```

```
for i in range(0,10,1):

    print(i)
```

12. Write a short program that prints the numbers 1 to 10 using a for loop. Then write an equivalent program that prints the numbers 1 to 10 using a while loop.

solution

#using for loop

```
for i in range(1,10):

    print(i)
```

```
# using while loop

i=1

while (i<=10):

    print(i)

    i+=1
```

13. If you had a function named bacon() inside a module named spam, how would you call it after importing spam?

solution

import spam

spam.bacon()