**Discussion 1**

What is the Python data type for each literal/expression below? Moreover, what is the resulting data value for each expression?

**1 / 2**

**1 // 2**

**1 > 2`**

**True**

**"True"**

**3 < 3 and 3/0 > 0**

**3/0 > 3 and 3 < 0**

**3 > 0 or 3/0 != 1**

**5 < 6 > 3**

**Discussion 2**

Consider the following Python program, for each line, from line1 to line6, state the output and briefly explain the reason.

a, b, c = 9, 10, 11

print (5>a or b<8 or 20>c>9) #line1

a, b, c = 4, -1, 11

c, b, a = a//2, int(c/3), b+c

print(a,b,c) #line2

a = 8 + 9

if a > 8:

print ("bigger") #line3

print("small") #line4

if a < 9:

print ("bigger") #line5

print("small") #line6

**Discussion 3**

**What is the output of the following program?**

# Define the conditions

is\_admin = True

is\_logged\_in = True

has\_permission = False

# Define the boolean expression for access control

access\_granted = is\_logged\_in and (is\_admin or has\_permission)

# Print the evaluation of the boolean expression

if access\_granted:

print("Access granted.")

else:

print("Access denied.")

**Discussion 4**

Real-Life Example: Smart Home Security System

A smart home security system uses boolean logic to determine whether to trigger an alarm based on various sensors and conditions. The system might include motion detectors, door/window sensors, and a time-based arming schedule.

A time-based arming schedule is a feature in security systems where the system automatically arms and disarms itself at predefined times. This scheduling ensures that the security system is active when needed and inactive when it’s not necessary, such as when people are home and moving around freely.

**Based on the Scenario** that should trigger an alarm if:

1. Motion is detected AND
2. The system is armed AND
3. A door or window is open

**Write the Boolean expression to c**heck if the alarm should be triggered**.**

**Discussion 5**

What is the output of the following program?

# Define user roles and request content

user\_role = "admin"

request\_content = "delete record"

# Define conditions for access

is\_admin = (user\_role == "admin")

is\_manager = (user\_role == "manager")

contains\_delete = ("delete" in request\_content)

contains\_record = ("record" in request\_content)

# Define the boolean expression for access control

access\_granted = (is\_admin or is\_manager) and contains\_delete and contains\_record

# Print the evaluation of the boolean expression

if access\_granted:

print("Access granted.")

else:

print("Access denied.")

**Discussion 6**

What is the output of the following program?

# List of items to categorize

items = ["apple", "carrot", "durian","tomato","banana", "spinach", "orange", "broccoli"]

# List of known fruits and vegetables

known\_fruits = ["apple", "banana", "orange"]

known\_vegetables = ["carrot", "spinach", "broccoli"]

num\_fruit=num\_veg=0

# Read each item and categorize it

for item in items:

if item in known\_fruits:

num\_fruit=num\_fruit+1

elif item in known\_vegetables:

num\_veg = num\_veg+1

else:

print(f"Item '{item}' is unknown and will not be categorized.")

# Output the categorized lists

print("Fruits:", num\_fruit)

print("Vegetables:", num\_veg)

**Discussion 7 (optional)**

Real-Life Example: Simple ATM Transaction

Let's consider an ATM allow a user to check their balance, deposit money, and withdraw money.

Partial code is given as follows:

# Initialize account balance

balance = 1000.0

while True:

print('''\nATM Menu:)

1. Check Balance

2. Deposit Money

3. Withdraw Money

4. Exit''')

# Get user choice

choice = int(input("Enter your choice (1-4): "))

Please complete the code based on the choices provided by the ATM.

**Choice 1**: Check Balance

* Displays the current balance.

**Choice 2**: Deposit Money

* Prompts the user to enter a deposit amount.
* Checks if the deposit amount is positive.
* Adds the deposit amount to the balance and displays the new balance.
* Displays an error message if the deposit amount is invalid.

**Choice 3**: Withdraw Money

* Prompts the user to enter a withdrawal amount.
* Checks if the withdrawal amount is positive and does not exceed the balance.
* Subtracts the withdrawal amount from the balance and displays the new balance.
* Displays an error message if the withdrawal amount exceeds the balance or is invalid.

**Choice 4**: Exit

* Exits the program and thanks the user.

**Invalid Choice**:

* Displays an error message if the user enters an invalid choice.