Lab 1: Expressions, Variables, and Data Types (Lecture 1)

Task 1: Exploring Python's Built-in Types

Use the Python console to determine the data type of the following values using the built-in **type()** function.

Value	Predicted Type	Command to Check	Actual Result
1234		type(1234)	
8.99		type(8.99)	
9.0		type(9.0)	
True		type(True)	
'1'		type('1')	
str(1)		type(str(1))	

Task 2: Arithmetic and Operator Precedence

Evaluate the following arithmetic expressions. Predict the final value and the resulting data type before entering the command.

1. What is the final result of: (13 - 4) / (12 * 12)
2. Compare the results and types:
   - type(4 * 3)
   - type(4.0 * 3)
3. Test integer division behavior: int(1 / 2)
4. Determine the final value of x:

   Python

   ```python
   x = 5
   # What is the result of 3*x - 5?
   # What is the result of 3 * (x-5)?
   # What is the current value of x?
   ```

Task 3: Assignment Rules and Variable Binding

1. **Allowed vs. Not Allowed:** Which of these lines execute without an error? For those that fail, identify the syntax rule that was violated.
   - x = 6
   - 6 = x (Why does this fail? What is an l-value?)
   - x*y = 3+4

- xy = 3+4

2. **Re-assignment Check:** Follow the steps below and check the variable values at each step.

   Python

   ```
   pi = 3.14
   radius = 2.2
   area = pi * (radius ** 2)

   radius = radius + 1
   area_new = pi * (radius ** 2)
   ```

   - Does changing radius automatically update the original area? Why or why not?

Lab 2: Type Conversion, Strings, and Logic (Lecture 2)

Task 4: Type Conversion, Rounding, and Built-in Functions

Perform the following conversions and function calls.

1. **Conversions:** Find the result of: float(123), int(7.2), int('1')
2. **Rounding:**
   - round(9.49)
   - round(9.50)
3. **Utility Functions:**
   - abs(-5) and abs(5)
   - len("Yo") and len("HoHo")
4. **Combined Operations:** Find the final result of: float(round(7.2))

Task 5: Mathematical Challenges (No Control Flow)

Goal: Use the math module and complex expressions to solve engineering problems *without* using if statements or loops. You must start by running: import math.

1. **Quadratic Roots:** Define the coefficients , , and

   .

   - Calculate the discriminant: discriminant = (b**2) - (4 * a * c)

- Calculate the two roots (and   ) using the

  expression   .

- **Check:**   should be -2.0, and   should be -3.0.

Python

```
import math
a = 1
b = 5
c = 6
# Calculate and print root1
# Calculate and print root2
```

2. **Distance Formula:** Define four variables for two points: and

   .

   - Calculate the distance   using the expression:
   - **Check:** The result should be 5.0.

Task 6: String Concatenation and Repetition

Predict and check the values of s1 and s2.

Python

```
b = ":"
c = ")"
s1 = b + 2 * c
# Value of s1?

f = "a"
g = " b"
h = "3"
s2 = (f + g) * int(h)
# Value of s2?
```

Task 7: String Indexing and Slicing (YOU TRY IT!)

Define the following string and predict the output of the slicing operations.

Python

```
s = "ABC d3f ghi"
# What are the results of these commands?
s[3:len(s)-1]
s[4:0:-1]  # Hint: Slicing backwards - watch the start and end points
s[6:3]     # Why is the result empty?
```

Task 8: Combining Different Types (Predicting Errors)

For the following expressions, **predict** whether the operation will result in a numerical value, a string value, or a **TypeError**. Then, run the command to verify.

Command	Predicted Result	Predicted Type	Actual Result
int('34') + 56			
str(12) + ' red roses'			
'50' * 3			
80.0 + int('100')			
str(5.2) + 7			
2.0 * 'this'			

- **Discussion:** What is the general rule regarding the use of the + operator with strings and numbers?

Task 9: Relational and Logical Operators

Use the following variables to evaluate the comparison expressions.

Python

```python
x = 14
w = 0
v = 4
y = 2
z = "cilantro"
a = "dog"
```

1. **Simple Comparisons:** Find the value of the expressions:
   - (x <= 13)
   - (w != 0.4)
   - (v < 4.0)
   - **Invalid Comparison:** What happens when you try to compare different types? (y > "ab")
2. **String Comparisons:** Find the value of the expressions:
   - (z == "coriander")
   - (a < "cat")

Task 10: Complex Logical Precedence

Define the following variables and evaluate the compound logical expressions.

Remember the order of operations for logic: **not** **and** **or**.

Python

```
a = 10
b = 20
c = 5
```

1. print((a < b) and (b / c == 4))
2. print(not ((a > b) or (c * 2 < a)))
3. print(a > 5 or b < 10 and c == 5) (Evaluate using precedence rules)
4. **Challenge:** print(not (a == 10) or (b < 15 and c > 0))