**Python Basics to Intermediate Student Handbook: Examples with Solutions**

**1. Variables and Data Types**

**Example 1: Variables and Simple Data Types**

**Task:**

Create a variable for each basic data type (int, float, string, and boolean) and print their values.

**Solution:**

# Variable declarations

integer\_variable = 10

float\_variable = 3.14

string\_variable = "Hello, World!"

boolean\_variable = True

# Printing values

print("Integer:", integer\_variable)

print("Float:", float\_variable)

print("String:", string\_variable)

print("Boolean:", boolean\_variable)

**Explanation**:

* integer\_variable stores an integer (10).
* float\_variable stores a floating-point number (3.14).
* string\_variable stores a string ("Hello, World!").
* boolean\_variable stores a boolean value (True).

**2. Arithmetic Operations**

**Example 2: Simple Arithmetic Operations**

**Task:**

Perform addition, subtraction, multiplication, division, and exponentiation on two numbers.

**Solution:**

# Define two numbers

a = 8

b = 4

# Perform arithmetic operations

sum\_result = a + b

sub\_result = a - b

prod\_result = a \* b

div\_result = a / b

exp\_result = a \*\* b

# Output results

print("Sum:", sum\_result)

print("Difference:", sub\_result)

print("Product:", prod\_result)

print("Division:", div\_result)

print("Exponentiation:", exp\_result)

**Explanation**:

* a + b adds a and b.
* a - b subtracts b from a.
* a \* b multiplies a and b.
* a / b divides a by b.
* a \*\* b raises a to the power of b.

**3. Conditional Statements**

**Example 3: If-Else Statements**

**Task:**

Write a program that checks whether a number is positive, negative, or zero.

**Solution:**

# Input a number

num = int(input("Enter a number: "))

# Check if the number is positive, negative, or zero

if num > 0:

print("The number is positive.")

elif num < 0:

print("The number is negative.")

else:

print("The number is zero.")

**Explanation**:

* if num > 0: Checks if the number is positive.
* elif num < 0: Checks if the number is negative.
* else: If the number is neither positive nor negative, it is zero.

**4. Loops**

**Example 4: For Loop**

**Task:**

Write a program to print numbers from 1 to 5 using a for loop.

**Solution:**

# Loop to print numbers from 1 to 5

for i in range(1, 6):

print(i)

**Explanation**:

* range(1, 6) generates a sequence of numbers from 1 to 5.
* The for loop iterates over each number in that range and prints it.

**Example 5: While Loop**

**Task:**

Write a program that counts down from 5 to 1 using a while loop.

**Solution:**

# Initialize counter

counter = 5

# Countdown using while loop

while counter > 0:

print(counter)

counter -= 1 # Decrease the counter by 1

**Explanation**:

* The while loop continues as long as the condition counter > 0 is true.
* The counter is decremented by 1 in each iteration.

**5. Functions**

**Example 6: Defining and Calling Functions**

**Task:**

Define a function that takes two arguments and returns their sum.

**Solution:**

# Function to add two numbers

def add\_numbers(x, y):

return x + y

# Call the function with arguments 3 and 5

result = add\_numbers(3, 5)

print("The sum is:", result)

**Explanation**:

* add\_numbers is a function that takes two parameters (x and y) and returns their sum.
* The function is called with the values 3 and 5, and the result is printed.

**6. Lists**

**Example 7: Lists and List Operations**

**Task:**

Create a list of numbers and perform some basic list operations (access elements, append, and loop through it).

**Solution:**

# Create a list

numbers = [1, 2, 3, 4, 5]

# Access an element by index

print("First number:", numbers[0])

# Append a new number to the list

numbers.append(6)

print("List after appending 6:", numbers)

# Loop through the list and print each number

for num in numbers:

print(num)

**Explanation**:

* numbers[0] accesses the first element of the list.
* numbers.append(6) adds the number 6 to the end of the list.
* The for loop iterates through each element in the list and prints it.

**7. Dictionaries**

**Example 8: Dictionaries and Accessing Values**

**Task:**

Create a dictionary of a person’s information and access its values using keys.

**Solution:**

# Create a dictionary

person = {"name": "Alice", "age": 25, "city": "New York"}

# Access values by keys

print("Name:", person["name"])

print("Age:", person["age"])

print("City:", person["city"])

**Explanation**:

* A dictionary person is created with keys "name", "age", and "city".
* Values can be accessed using the keys, e.g., person["name"].

**8. Error Handling**

**Example 9: Try-Except Block**

**Task:**

Write a program that catches a division by zero error.

**Solution:**

# Try to divide by zero

try:

result = 10 / 0

except ZeroDivisionError:

print("Error: Division by zero is not allowed.")

**Explanation**:

* The try block contains the code that may cause an error (division by zero in this case).
* The except block catches the ZeroDivisionError and prints an error message.

**9. File Handling**

**Example 10: Reading from a File**

**Task:**

Write a program that reads the content of a text file and prints it.

**Solution:**

python

Copy

# Open the file for reading

with open("sample.txt", "r") as file:

content = file.read()

print(content)

**Explanation**:

* The with open() statement opens the file "sample.txt" in read mode ("r").
* The file.read() method reads the entire content of the file and stores it in the variable content.