Complete Python Notes – Prepared by Prince

Modern Codebook Style with Colored Code Blocks

Python Basics

• Introduction, Features, Installation, Comments, Keywords
• Variables, Constants, Data Types, Type Casting
• Input and Output Functions

Operators

• Arithmetic, Relational, Logical, Assignment, Bitwise
• Membership (in, not in) and Identity (is, is not)
• Operator Precedence

Control Flow

• if, elif, else
• for and while loops
• break, continue, pass

Lists and Methods

• append(), extend(), insert(), pop(), remove(), clear(), sort(), reverse(), count(), index()

Tuples

• Immutable lists
• Methods: count(), index()

Sets

• Unique unordered collection
• Methods: add(), update(), remove(), discard(), pop(), clear()

Dictionaries

• Key-value pairs
• Methods: get(), keys(), values(), items(), update(), pop(), popitem(), clear()

Functions

• Defining and Calling Functions
• *args, **kwargs, Lambda & Recursion

OOP Concepts

• Class, Object, Constructor (__init__)
• Inheritance, Polymorphism, Encapsulation, Abstraction, Singleton

File Handling

• Open, Read, Write, Append, File modes: r, w, a, r+

Garbage Collection

• Automatic memory management using gc module

Advanced Topics

• Iterators and Generators
• Decorators
• Context Managers
• Regular Expressions
• Multithreading and Multiprocessing

Data Science Libraries

• NumPy – numerical computing
• Pandas – data analysis
• Matplotlib / Seaborn – data visualization
• Scikit-learn – machine learning
• Streamlit – web apps

Example: Armstrong Number

```
num = 153 sum = 0 temp = num while temp > 0: digit = temp % 10 sum += digit ** 3 temp //= 10 if num ==
sum: print("Armstrong number") else: print("Not an Armstrong number")
```

Example: Palindrome Check

```
string = "madam" if string == string[::-1]: print("Palindrome") else: print("Not Palindrome")
```

Example: Fibonacci Series

```
a, b = 0, 1 for i in range(10): print(a, end=" ") a, b = b, a + b print()
```

Example: Prime Number Check

```
num = 17 if num > 1: for i in range(2, num): if num % i == 0: print("Not Prime") break else:
print("Prime")
```

Example: Factorial Program

```
def factorial(n): if n == 0 or n == 1: return 1 else: return n * factorial(n-1) print(factorial(5))
```

Example: Reverse a Number

```
num = 12345 rev = 0 temp = num while temp > 0: rev = rev*10 + temp%10 temp //= 10 print(rev)
```

Example: Swapping Two Numbers

```
a, b = 5, 10 a, b = b, a print(a, b)
```

Example: Even/Odd Check

```
num = 7 if num % 2 == 0: print("Even") else: print("Odd")
```

Example: Sum of Digits

```
num = 123 sum = 0 temp = num while temp > 0: sum += temp % 10 temp //= 10 print(sum)
```

Example: Largest of Three Numbers

```
a, b, c = 5, 10, 7 largest = max(a, b, c) print(largest)
```

Example: List Comprehension

```
numbers = [1, 2, 3, 4, 5] squares = [x**2 for x in numbers] print(squares)
```

Example: Map, Filter, Reduce

```
from functools import reduce numbers = [1, 2, 3, 4, 5] squared = list(map(lambda x: x**2, numbers))
even = list(filter(lambda x: x%2==0, numbers)) sum_all = reduce(lambda x, y: x+y, numbers)
print(squared, even, sum_all)
```

Example: *args and **kwargs Example

```
def func(*args, **kwargs): print(args) print(kwargs) func(1, 2, 3, name='Prince', age=25)
```

Example: Exception Handling Example

```
try: x = 1/0 except ZeroDivisionError: print("Cannot divide by zero") finally: print("Execution Complete")
```

Example: File Handling Example

```
with open('sample.txt', 'w') as f: f.write("Hello Python") with open('sample.txt', 'r') as f:
print(f.read())
```

Example: Singleton Example

```
class Singleton: _instance = None def __new__(cls): if cls._instance is None: cls._instance = super().__new__(cls) return cls._instance a = Singleton() b = Singleton() print(a is b) # True
```

Example: Garbage Collection Example

```
import gc print(gc.isenabled()) gc.collect()
```