$ ASK TEACHER HOW TO WRITE THEN WRITE

$ Add output of each program >> or ask me for output

```
# Arithmetic Operations
num1 = 10
num2 = 5
addition = num1 + num2
subtraction = num1 - num2
multiplication = num1 * num2
division = num1 / num2
modulus = num1 % num2
print("Addition:", addition)
print("Subtraction:", subtraction)
print("Multiplication:", multiplication)
print("Division:", division)
print("Modulus:", modulus)
# String Operations
message = "Hello, "
name = "John"
full_message = message + name
print(full_message)
# Input/Output
user_input = input("Enter your name: ")
print("Hello, " + user_input)
```

2. Control Flow Constructs: If-else, Relational and Logical Operators

```python
# If-Else Statements
num = int(input("Enter a number: "))
if num > 0:
 print("Positive number.")
elif num == 0:
 print("Zero.")
else:
 print("Negative number.")
# Relational and Logical Operators
x = 5
y = 10
print(x < y)
print(x >= y)
print(x == y)
print(x != y)
```

```
# Using Logical Operators
a = True
b = False
print(a and b)
print(a or b)
print(not a)
```
3. Iteration: While loop, for loop
```python
# While Loop
count = 1
while count <= 5:
 print("Count:", count)
 count += 1
# For Loop
numbers = [1, 2, 3, 4, 5]
for num in numbers:
 print("Number:", num)

```

4. Collections: Lists, Tuples
```python
# Lists
fruits = ["apple", "banana", "orange"]
print(fruits[0]) # Accessing element
fruits.append("grape") # Adding element
fruits.remove("banana") # Removing element
```

```
print(fruits)
# Tuples
coordinates = (10, 20)
print("X-coordinate:", coordinates[0])
print("Y-coordinate:", coordinates[1])
```
5. Collections: Sets, Dictionary
```python
# Sets
fruits_set = {"apple", "banana", "orange", "apple"}
print(fruits_set) # Duplicate elements are automatically removed
fruits_set.add("grape") # Adding element
fruits_set.remove("banana") # Removing element
print(fruits_set)
# Dictionary
student = {
 "name": "John",
 "age": 20,
 "course": "Computer Science"
}
print(student["name"])
print(student.get("age"))
student["age"] = 21 # Updating value
print(student)
```
6. Functions and Modules: Sys, Math, Time
```python
# Sys Module
import sys
print(sys.platform)
# Math Module
import math
print(math.sqrt(25))
print(math.pi)
# Time Module
import time
print("Current Time:", time.strftime("%H:%M:%S"))
```
7. File Handling: Data streams, Access modes, Read/Write/Seek
```python
# Writing to a File
with open("data.txt", "w") as file:
 file.write("Hello, World!\n")
 file.write("This is a text file.")
```

```
# Reading from a File
with open("data.txt", "r") as file:
 content = file.read()
 print(content)
```
8. OOP's, Classes, Objects, Exception Handling
```python
# Classes and Objects
class Circle:
 def __init__(self, radius):
 self.radius = radius
 def area(self):
 return 3.14 * self.radius ** 2
circle1 = Circle(5)
print("Area of Circle:", circle1.area())
# Exception Handling
try:
 num1 = int(input("Enter a number: "))
 num2 = int(input("Enter another number: "))
 result = num1 / num2
 print("Result:", result)
except ZeroDivisionError:
 print("Cannot divide by zero.")
except ValueError:
 print("Invalid input. Please enter a number.")
```
9. GUI programming: TkInter
```python
import tkinter as tk
def display_message():
 message = "Hello, " + entry.get()
 label.config(text=message)
root = tk.Tk()
root.title("Greeting App")
label = tk.Label(root, text="Enter your name:")
label.pack()
entry = tk.Entry(root)
entry.pack()
button = tk.Button(root, text="Greet", command=display_message)
button.pack()
root.mainloop()
```