

--- Conditional Statements Section ---

if-else statement (guessing student category)

```
print("Tell me your age?")
myage = int(input())
if myage < 30:
    print("Welcome to the Club.")
else:
    print("Oh! No. You are not accepted.")
```

if-elif-else statement (grade assignment)

```
score = int(input("Enter your score: "))
if score >= 90:
    grade = 'A'
elif score >= 80:
    grade = 'B'
elif score >= 70:
    grade = 'C'
elif score >= 60:
    grade = 'D'
else:
    grade = 'F'
print(f"Your grade is: {grade}")
```

--- Loops Section ---

for loop (multiplication table)

```
n = int(input("구구단 몇 단을 계산할까요? "))
print(f"구구단 {n}단을 계산합니다.")
for i in range(1, 10):
    print(f"{n} x {i} = {n * i}")
```

while loop (number guessing game)

```
import random
guess_number = random.randint(1, 100)
print("숫자를 맞춰 보세요. (1 ~ 100)")
user_input = int(input())
while user_input != guess_number:
    if user_input > guess_number:
        print("숫자가 너무 큼니다.")
    else:
```

```
        print("숫자가 너무 작습니다.")
    user_input = int(input())
print(f"정답입니다. 입력한 숫자는 {user_input}입니다.")
```

for loop (range with steps)

```
for i in range(1, 10, 2): # increments by 2
    print(i)
```

--- Functions Section ---

Simple function (hello world)

```
def print_hello_world():
    print("Hello World")
```

```
print_hello_world()
```

Function with parameters (rectangle area)

```
def calculate_rectangle_area(x, y):
    return x * y
```

```
x = 5
```

```
y = 7
```

```
print(f"The area of the rectangle is: {calculate_rectangle_area(x, y)}")
```

Recursive function (factorial)

```
def factorial(n):
    if n == 1:
        return 1
    else:
        return n * factorial(n - 1)
```

```
num = int(input("Enter a number for factorial calculation: "))
```

```
print(f"Factorial result: {factorial(num)}")
```

Function with default arguments

```
def greet(name, msg="Hello"):
    print(f"{msg}, {name}!")
```

```
greet("Sungchul")
```

```
greet("Sungchul", "Good Morning")
```

--- Error Handling and Debugging Section ---

Handling indentation error example

```
def example_indentation():
```

```
    x = 5
```

```
    print("This is properly indented:", x)
```

```
example_indentation()
```

--- Advanced Topics: Variable-length Arguments ---

Variable-length arguments (*args)

```
def asterisk_test(a, b, *args):
```

```
    return a + b + sum(args)
```

```
print(asterisk_test(1, 2, 3, 4, 5)) # 15
```

Keyword arguments (**kwargs)

```
def kwargs_test(**kwargs):
```

```
    for key, value in kwargs.items():
```

```
        print(f"{key}: {value}")
```

```
kwargs_test(first=3, second=4, third=5)
```

--- Scoping Rules Section ---

Local vs Global Variables

```
x = 10 # global variable
```

```
def local_vs_global():
```

```
    global x
```

```
    x = 20 # modifies the global variable
```

```
    print(f"Inside function, x: {x}")
```

```
local_vs_global()
```

```
print(f"Outside function, x: {x}")
```

--- Additional Labs ---

Reversing a string

```
sentence = "I love you"
```

```
reverse_sentence = ""
```

```
for char in sentence:
```

```
    reverse_sentence = char + reverse_sentence
```

```
print(f"Reversed sentence: {reverse_sentence}")
```

Decimal to binary conversion

```
decimal = 10
```

```
binary_result = ""
```

```
while decimal > 0:
```

```
    remainder = decimal % 2
```

```
    decimal = decimal // 2
```

```
    binary_result = str(remainder) + binary_result
```

```
print(f"Binary representation of decimal 10: {binary_result}")
```

Average calculation from 2D list

```
kor_score = [49, 80, 20, 100, 80]
```

```
math_score = [43, 60, 85, 30, 90]
```

```
eng_score = [49, 82, 48, 50, 100]
```

```
midterm_score = [kor_score, math_score, eng_score]
```

```
student_score = [0, 0, 0, 0, 0]
```

```
i = 0
```

```
for subject in midterm_score:
```

```
    for score in subject:
```

```
        student_score[i] += score
```

```
    i += 1
```

```
i = 0
```

```
a, b, c, d, e = student_score
```

```
student_average = [a/3, b/3, c/3, d/3, e/3]
```

```
print(f"Student average scores: {student_average}")
```

--- Final Project: Trapezium Area Calculation ---

Function to calculate trapezium area

```
def trapezium_area(base, top, height):
```

```
    return ((base + top) * height) / 2
```

```
base = float(input("Enter base length: "))
```

```
top = float(input("Enter top length: "))
```

```
height = float(input("Enter height: "))
```

```
print(f"Trapezium area: {trapezium_area(base, top, height)}")
```