

ASSIGNMENT:3.3

Name: SANIYA

Roll no: 2403A510E7

Branch:CSE

Batch:05

Task Description#1

- Try 3 different prompts to generate a factorial function.

Expected Output#1

- Comparison of AI-generated code styles

Prompt-1:

Generate a function that prints factorial of a given number

Code:

ASSIGNMENT:3.3

```
task1.py > main
1 def factorial(n):
2     if n < 0:
3         return "Factorial is not defined for negative numbers"
4     elif n == 0 or n == 1:
5         return 1
6     else:
7         result = 1
8         for i in range(2, n + 1):
9             result *= i
10        return result
11
12 def main():
13     Ctrl+L to chat, Ctrl+K to generate
14     try:
15         # Get input from user
16         user_input = input("Enter a number to calculate factorial: ")
17
18         # Convert to integer
19         number = int(user_input)
20
21         # Calculate factorial
22         result = factorial(number)
23
24         # Print the result
25         if isinstance(result, str):
26             print(result)
27         else:
28             print(f"Factorial of {number} is: {result}")
29
30     except ValueError:
31         print("Error: Please enter a valid integer")
32     except Exception as e:
33         print(f"An error occurred: {e}")
34
35 if __name__ == "__main__":
36     main()
37
```

ASSIGNMENT:3.3

Output:

```
Enter a number to calculate factorial: 5
Factorial of 5 is: 120
PS C:\Users\thoop\OneDrive\文档\AIAC_LAB_Lab3> & C:/Users/thoop/AppData/Local/Programs/Python/Python313/python.exe c:/Users/thoop/OneDrive/文档/AIAC_LAB_Lab3/task1.py
Enter a number to calculate factorial: 8
Factorial of 8 is: 40320
```

Prompt 2:

Write a simple python function to find factorial of a number ,collect input from the console

Code:

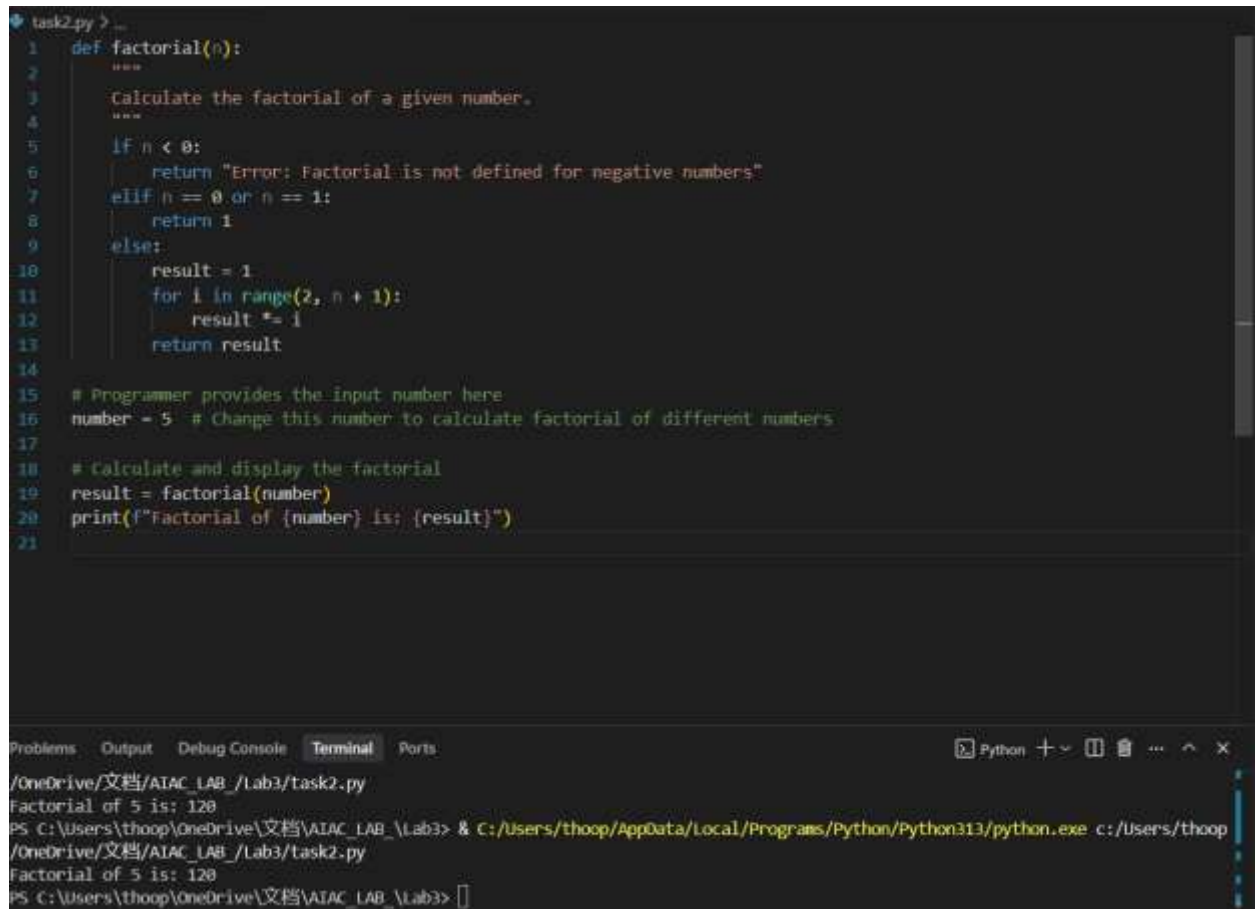
```
task2.py > ...
1  def factorial(n):
2      if n < 0:
3          return "Error: Factorial is not defined for negative numbers"
4      elif n == 0 or n == 1:
5          return 1
6      else:
7          result = 1
8          for i in range(2, n + 1):
9              result *= i
10             return result
11
12 # Collect input from console
13 user_input = input("Enter a number to calculate factorial: ")
14
15 try:
16     number = int(user_input)
17     result = factorial(number)
18     print(f"Factorial of {number} is: {result}")
19 except ValueError:
20     print("Error: Please enter a valid integer")
21

Problems Output Debug Console Terminal Ports
Python + - [] ... ^ >
6
PS C:\Users\thoop\OneDrive\文档\AIAC_LAB_Lab3> & C:/Users/thoop/AppData/Local/Programs/Python/Python313/python.exe c:/Users/thoop/OneDrive/文档/AIAC_LAB_Lab3/task2.py
Enter a number to calculate factorial: 5
Factorial of 5 is: 120
PS C:\Users\thoop\OneDrive\文档\AIAC_LAB_Lab3> []
```

Prompt-3:

Create a simple python function that gives factorial of a number,use input from the programmer

ASSIGNMENT:3.3



```
task2.py > ...
1 def factorial(n):
2     """
3     calculate the factorial of a given number.
4     """
5     if n < 0:
6         return "Error: Factorial is not defined for negative numbers"
7     elif n == 0 or n == 1:
8         return 1
9     else:
10        result = 1
11        for i in range(2, n + 1):
12            result *= i
13        return result
14
15 # Programmer provides the input number here
16 number = 5 # Change this number to calculate factorial of different numbers
17
18 # Calculate and display the factorial
19 result = factorial(number)
20 print(f"Factorial of {number} is: {result}")
21
```

Problems Output Debug Console Terminal Ports

/OneDrive/文档/AIAC_LAB/Lab3/task2.py
Factorial of 5 is: 120
PS C:\Users\thoop\OneDrive\文档\AIAC_LAB\Lab3> & C:/Users/thoop/AppData/Local/Programs/Python/Python313/python.exe c:/Users/thoop/OneDrive/文档/AIAC_LAB/Lab3/task2.py
Factorial of 5 is: 120
PS C:\Users\thoop\OneDrive\文档\AIAC_LAB\Lab3> []

Task Description#2

- Provide a clear example input-output prompt to generate a sorting function.

Expected Output#2

- Functional sorting code from AI

Prompt:

Generate a python code that takes a input list from the user and sort the list
Ascending order

Code along with output:

ASSIGNMENT:3.3

```
1 def sort_list_ascending():
2     """
3     Function to take input list from user and sort it in ascending order.
4     """
5     try:
6         # Get input from user
7         print("Enter numbers separated by spaces (e.g., 5 2 8 1 9):")
8         user_input = input("Enter your list: ")
9
10        # Convert input string to list of integers
11        numbers = [int(x) for x in user_input.split()]
12
13        # Sort the list in ascending order
14        sorted_list = sorted(numbers)
15
16        # Display the results
17        print(f"Original list: {numbers}")
18        print(f"Sorted list (ascending): {sorted_list}")
19
20    except ValueError:
21        print("Error: Please enter valid numbers separated by spaces")
22    except Exception as e:
23        print(f"An error occurred: {e}")
24
25 def sort_list_with_builtin():
26     """
27     Alternative method using list.sort() method
28     """
29     try:
30         print("Enter numbers separated by spaces:")
31         user_input = input("Enter your list: ")
32
33         # Convert input string to list of integers
34         numbers = [int(x) for x in user_input.split()]
35
36         # Create a copy to avoid modifying original list
37         numbers_copy = numbers.copy()
38
39    def sort_list_with_builtin():
40
41        # Convert input string to list of integers
42        numbers = [int(x) for x in user_input.split()]
43
44        # Create a copy to avoid modifying original list
45        numbers_copy = numbers.copy()
46
47        # Sort the list in ascending order using sort() method
48        numbers_copy.sort()
49
50        print(f"Original list: {numbers}")
51        print(f"Sorted list (ascending): {numbers_copy}")
52
53    except ValueError:
54        print("Error: Please enter valid numbers separated by spaces")
55
56 # Main execution
57 if __name__ == "__main__":
58     print("=== List Sorting Program ===")
59     print("Method 1: Using sorted() function")
60     sort_list_ascending()
61
62     print("\n" + "-"*40 + "\n")
63     print("Method 2: Using sort() method")
64     sort_list_with_builtin()
65
66 Problems Output Debug Console Terminal Ports
```

Method 1: Using sorted() function
Enter numbers separated by spaces (e.g., 5 2 8 1 9):
Enter your list: 5 66 4 22 3 8 99 0 33
Original list: [5, 66, 4, 22, 3, 8, 99, 0, 33]
Sorted list (ascending): [0, 3, 4, 5, 8, 22, 33, 66, 99]

Task Description#3

- Start with the vague prompt “Generate python code to
- calculate power bill” and improve it step-by-step

ASSIGNMENT:3.3

prompt:

generate a python code to calculate powerbill containing previous units ,present units ,read input units from the console

code:

```
#!/usr/bin/env python3
def calculate_power_bill():
    """
    Function to calculate power bill based on previous and present units
    """
    try:
        # Get input from user
        print("\n--- Power Bill Calculator ---")

        # Get previous units
        previous_units = float(input("Enter previous month units: "))

        # Get present units
        present_units = float(input("Enter present month units: "))

        # Calculate units consumed
        units_consumed = present_units - previous_units

        # Calculate bill based on units consumed
        if units_consumed <= 0:
            print("Error: Present units should be greater than previous units")
            return

        # Define rate structure (example rates - can be modified)
        if units_consumed <= 100:
            rate_per_unit = 2.50
        elif units_consumed <= 200:
            rate_per_unit = 3.50
        elif units_consumed <= 300:
            rate_per_unit = 4.50
        else:
            rate_per_unit = 5.50

        # Calculate total bill
        total_bill = units_consumed * rate_per_unit

        # Display the results
        print("\n--- Bill Details ---")
        print(f"Previous month units: {previous_units}")
        print(f"Present month units: {present_units}")
        print(f"Units consumed: {units_consumed}")
        print(f"Rate per unit: ${rate_per_unit}")
        print(f"Total bill: ${total_bill:.2f}")

    except ValueError:
        print("Error: Please enter valid numbers")
    except Exception as e:
        print(f"An error occurred: {e}")

def calculate_power_bill_with_tax():
    """
    Function to calculate power bill with tax included
    """
    try:
        print("\n--- Power Bill Calculator with Tax ---")

        # Get input from user
        previous_units = float(input("Enter previous month units: "))
        present_units = float(input("Enter present month units: "))

        # Calculate units consumed
        units_consumed = present_units - previous_units

        # Calculate bill
        if units_consumed <= 0:
            print("Error: Present units should be greater than previous units")
            return

        # Calculate base bill
        if units_consumed <= 100:
            rate_per_unit = 2.50
        elif units_consumed <= 200:
            rate_per_unit = 3.50
        elif units_consumed <= 300:
            rate_per_unit = 4.50
        else:
            rate_per_unit = 5.50
```

ASSIGNMENT:3.3

```
73     rate_per_unit = 4.50
74 else:
75     rate_per_unit = 5.50
76
77 base_bill = units_consumed * rate_per_unit
78
79 # calculate tax (example: 10% tax)
80 tax_rate = 0.10
81 tax_amount = base_bill * tax_rate
82
83 # Calculate total bill with tax
84 total_bill = base_bill + tax_amount
85
86 # Display detailed results
87 print("\n== Detailed Bill ==")
88 print(f"Previous month units: {previous_units}")
89 print(f"Present month units: {present_units}")
90 print(f"Units consumed: {units_consumed}")
91 print(f"Rate per unit: ${rate_per_unit}")
92 print(f"Base bill: ${base_bill:.2f}")
93 print(f"Tax: {(tax_rate*100)}%: ${tax_amount:.2f}")
94 print(f"Total bill: ${total_bill:.2f}")
95
96 except ValueError:
97     print("Error: Please enter valid numbers")
98 except Exception as e:
99     print(f"An error occurred: {e}")
100
101 # Main execution
102 if __name__ == "__main__":
103     # Run basic power bill calculator
104     calculate_power_bill()
105
106     # Run power bill calculator with tax
107     calculate_power_bill_with_tax()
```

Python 3.10.4 Shell

```
== Bill Details ==
Previous month units: 100.0
Present month units: 200.0
Units consumed: 100.0
Rate per unit: $5.5
Base bill: $550.00
Tax: 10%: $55.00
Total bill: $605.00
```

Task Description#4

- Write structured comments to help AI generate two linked functions (e.g., login_user() and register_user()).

Prompt:

Generate simple two python linked functions one is for register user and login user for account creation

Code:

ASSIGNMENT:3.3

```
task4.py > _
1  # Global dictionary to store user accounts
2  user_accounts = {}
3
4  def register_user():
5      """
6      Function to register a new user account
7      """
8      print("=== Account Registration ===")
9
10     # Get user input for account creation
11     username = input("Enter username for your account: ").strip()
12
13     # Check if username already exists
14     if username in user_accounts:
15         print("Error: Username already exists! Please choose a different username.")
16         return False
17
18     # Get password for account
19     password = input("Enter password for your account: ")
20
21     # Store user account credentials
22     user_accounts[username] = password
23
24     print("Account created successfully!")
25     print(f"Username: {username}")
26     print("You can now login with your account.")
27     return True
28
29  def login_user():
30      """
31      Function to login to existing user account
32      """
33      print("=== Account Login ===")
34
35      # Get login credentials
36      username = input("Enter your username: ").strip()
37      password = input("Enter your password: ")
38
39      # Check if account exists and password matches
40      if username in user_accounts and user_accounts[username] == password:
41          print("Login successful!")
42          print(f"Welcome to your account, {username}!")
43          return True
44      else:
45          print("Invalid username or password.")
46          print("Please check your account credentials.")
47          return False
48
49  # Main execution for account management
50  if __name__ == "__main__":
51      while True:
52          print("\n=== Account Management System ===")
53          print("1. Create Account")
54          print("2. Login to Account")
55          print("3. Exit")
56
57          choice = input("Enter your choice (1-3): ").strip()
58
59          if choice == '1':
60              register_user()
61          elif choice == '2':
62              login_user()
63          elif choice == '3':
64              print("Thank you for using Account Management System!")
65              break
66          else:
67              print("Invalid choice! Please try again.")
68
```

Output:

ASSIGNMENT:3.3

```
=== User Registration ===
Enter username: 2403a51057
Enter password: hary66
Confirm password: hary66
Enter email: 57@gmail.com
Enter full name: HarichandanaThoopukari
Enter phone number: 1236661230

=== Registration Successful ===
Username: 2403a51057
Email: 57@gmail.com
Full Name: HarichandanaThoopukari
Phone: 1236661230
Registration Date: 2024-01-01
Status: Active

=== User Management System ===
1. Register User
2. Login User
3. Display All Users
4. Exit

Enter your choice (1-4): 2
=== User Login ===
Enter username: 2403a51057
Enter password: hary66

=== Login Successful ===
Welcome back, HarichandanaThoopukari!
Username: 2403a51057
Email: 57@gmail.com
Full Name: HarichandanaThoopukari
Phone: 1236661230
Registration Date: 2024-01-01
Status: Logged In
Last Login: Current Session
```

Task Description#5

- Analyzing Prompt Specificity: Improving Temperature Conversion Function with Clear Instructions

Prompt1:

Write a Python function to convert temperature from Celsius to Fahrenheit or vice-versa as for the user`s selection

Code:

ASSIGNMENT:3.3

```
task5.py > ...
1  def convert_temperature():
2      """
3      Converts temperature between Celsius and Fahrenheit based on user input.
4      """
5      try:
6          print("Temperature Converter")
7          temp = float(input("Enter the temperature value: "))
8          unit = input("Is this in Celsius or Fahrenheit? (C/F): ").strip().upper()
9
10         if unit == 'C':
11             converted = (temp * 9/5) + 32
12             print(f"{temp}°C is {converted:.2f}°F")
13         elif unit == 'F':
14             converted = (temp - 32) * 5/9
15             print(f"{temp}°F is {converted:.2f}°C")
16         else:
17             print("Invalid unit. Please enter 'C' for Celsius or 'F' for Fahrenheit.")
18     except ValueError:
19         print("Invalid input. Please enter a numeric temperature value.")
20
21 # Call the function if this script is run directly
22 if __name__ == "__main__":
23     convert_temperature()
24
```

```
Problems  Output  Debug Console  Terminal  Ports
p' '--' 'c:\Users\thoop\OneDrive\文档\AIAC_LAB_\Lab3\task5.py'
Temperature Converter
Enter the temperature value: 15
Is this in Celsius or Fahrenheit? (C/F): c
15.0°C is 59.00°F
PS C:\Users\thoop\OneDrive\文档\AIAC_LAB_\Lab3> |
```

Prompt2:

Write a Python function to convert temperature from Celsius to Fahrenheit or vice-versa as for the user's selection.

Code along with output:

ASSIGNMENT:3.3

```
task5_1.py > ...
1 def convert_temperature():
2     """
3     Converts temperature between Celsius and Fahrenheit based on user's selection.
4     """
5     try:
6         print("Temperature Converter")
7         print("Select conversion type:")
8         print("1. Celsius to Fahrenheit")
9         print("2. Fahrenheit to Celsius")
10        choice = input("Enter 1 or 2: ").strip()
11
12        if choice == '1':
13            celsius = float(input("Enter temperature in Celsius: "))
14            fahrenheit = (celsius * 9/5) + 32
15            print(f"{celsius}°C is {fahrenheit:.2f}°F")
16        elif choice == '2':
17            fahrenheit = float(input("Enter temperature in Fahrenheit: "))
18            celsius = (fahrenheit - 32) * 5/9
19            print(f"{fahrenheit}°F is {celsius:.2f}°C")
20        else:
21            print("Invalid selection. Please enter 1 or 2.")
22    except ValueError:
23        print("Invalid input. Please enter a numeric temperature value.")
24
25 if __name__ == "__main__":
26     convert_temperature()
```

Problems Output Debug Console Terminal Ports

Select conversion type:
1. Celsius to Fahrenheit
2. Fahrenheit to Celsius
Enter 1 or 2: 1
Enter temperature in Celsius: 35
35.0°C is 95.00°F
PS C:\Users\thoop\OneDrive\文档\AIAC_LAB_\Lab3> |