

ASSIGNMENT-3.2

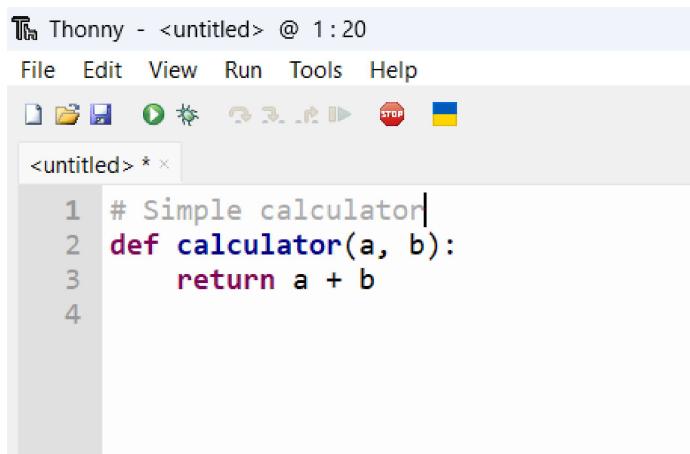
NAME:-K Ujwal

BATCH-16

HTNO:-2303A51058

Task 1: Progressive Prompting for Calculator Design

Initial AI Output (Basic Calculator)



The screenshot shows the Thonny Python IDE interface. The title bar says "Thonny - <untitled> @ 1:20". The menu bar includes File, Edit, View, Run, Tools, and Help. Below the menu is a toolbar with icons for file operations like Open, Save, and Run. The code editor window contains a single file named "<untitled>" with the following content:

```
1 # Simple calculator
2 def calculator(a, b):
3     return a + b
4
```

Improved AI Output (With Comments & Usage Examples):-

The screenshot shows the Thonny IDE interface. The top window is titled "Thonny - <untitled> @ 27:1". The menu bar includes File, Edit, View, Run, Tools, and Help. Below the menu is a toolbar with icons for file operations. The main code editor window contains an untitled script with the following Python code:

```
1 # Calculator program using functions
2 # Supports addition, subtraction, multiplication, and division
3
4 def add(a, b):
5     return a + b
6
7 def subtract(a, b):
8     return a - b
9
10 def multiply(a, b):
11     return a * b
12
13 def divide(a, b):
14     if b == 0:
15         return "Division by zero not allowed"
16     return a / b
17
18
19 # Example usage
20 x = int(input("Enter first number: "))
21 y = int(input("Enter second number: "))
22
23 print("Addition:", add(x, y))
24 print("Subtraction:", subtract(x, y))
25 print("Multiplication:", multiply(x, y))
26 print("Division:", divide(x, y))
27
```

The bottom window is titled "Shell". It displays the output of running the script. The user enters two numbers, 4 and 9, and the program prints the results of addition, subtraction, multiplication, and division:

```
Enter first number: 4
Enter second number: 9
Addition: 13
Subtraction: -5
Multiplication: 36
Division: 0.4444444444444444
```

Task 2: Refining Prompts for Sorting Logic

Initial AI Output (Vague Prompt):-

Thonny - <untitled> @ 6:1

File Edit View Run Tools Help

<untitled> * ×

```
1 # Sort student marks
2
3 marks = [78, 45, 89, 60, 92]
4 marks.sort()
5 print(marks)
6
```

Shell ×

```
>>> %Run -c $EDITOR_CONTENT
[45, 60, 78, 89, 92]
>>>
```

Refined AI Output (Clear Prompt with Order & Constraints):-

The screenshot shows the Thonny Python IDE interface. The top menu bar includes File, Edit, View, Run, Tools, and Help. Below the menu is a toolbar with icons for file operations like Open, Save, and Run. The main code editor window has a tab labeled <untitled>. The code itself is:

```
1 # Sort student marks in descending order
2
3 marks = [78, 45, 89, 60, 92]
4
5 sorted_marks = sorted(marks, reverse=True)
6
7 print("Sorted marks (Descending):", sorted_marks)
8
```

Below the code editor is a shell window titled "Shell". It displays the command ">>> %Run -c \$EDITOR_CONTENT" followed by the output "Sorted marks (Descending): [92, 89, 78, 60, 45]" and a final ">>>".

Task 3: Few-Shot Prompting for Prime Number Validation

AI-Generated Prime Checking Function:-

Thonny - <untitled> @ 19:1

File Edit View Run Tools Help

<untitled> *

```
1 # Function to check whether a number is prime
2
3 def is_prime(num):
4     if num <= 1:
5         return False
6
7     for i in range(2, int(num ** 0.5) + 1):
8         if num % i == 0:
9             return False
10
11    return True
12
13
14 # Test cases
15 print(is_prime(2))
16 print(is_prime(7))
17 print(is_prime(10))
18 print(is_prime(1))
19
```

Shell

```
>>> %Run -c $EDITOR_CONTENT
True
True
False
False
>>>
```

Task 4: Prompt-Guided UI Design for Student Grading System:-

Thonny - <untitled> @ 27:1

File Edit View Run Tools Help

<untitled> * ×

```
1 # Student grading system
2
3 name = input("Enter student name: ")
4 marks1 = int(input("Enter marks in Subject 1: "))
5 marks2 = int(input("Enter marks in Subject 2: "))
6 marks3 = int(input("Enter marks in Subject 3: "))
7
8 total = marks1 + marks2 + marks3
9 percentage = total / 3
10
11 if percentage >= 90:
12     grade = "A"
13 elif percentage >= 75:
14     grade = "B"
15 elif percentage >= 60:
16     grade = "C"
17 elif percentage >= 40:
18     grade = "D"
19 else:
20     grade = "Fail"
21
22 print("\n--- Student Report ---")
23 print("Name:", name)
24 print("Total Marks:", total)
25 print("Percentage:", percentage)
26 print("Grade:", grade)
27
```

Shell ×

```
>>> %Run -c $EDITOR_CONTENT

Enter student name: ROHAN
Enter marks in Subject 1: 87
Enter marks in Subject 2: 96
Enter marks in Subject 3: 98

--- Student Report ---
Name: ROHAN
Total Marks: 281
Percentage: 93.66666666666667
Grade: A

>>> |
```

Task 5: Prompt Specificity in Unit Conversion Functions:-

Thonny - <untitled> @ 13:1

File Edit View Run Tools Help



<untitled> * ×

```
1 # Unit conversion functions
2
3 def km_to_miles(km):
4     return km * 0.621371
5
6 def miles_to_km(miles):
7     return miles / 0.621371
8
9
10 # Test cases
11 print("10 km =", km_to_miles(10), "miles")
12 print("5 miles =", miles_to_km(5), "km")
13
```

Shell ×

```
>>> %Run -c $EDITOR_CONTENT
10 km = 6.21371 miles
5 miles = 8.046722489462816 km
>>>
```