Assignment 6.1

# TASK 1:

Use AI to create an Employee class with attributes (name,  
id, salary) and a method to calculate yearly salary.  
• Instructions:  
o Prompt AI to generate the Employee class.  
o Analyze the generated code for correctness and structure.  
o Ask AI to add a method to give a bonus and recalculate  
salary

PROMPT :

create Employee class with attributes (name, id, salary) and a method to calculate yearly salary and add a method to give bouns and recalculate the salary

CODE AND OUTPUT :

A screenshot of a computer

AI-generated content may be incorrect.

TASK 2 :

Prompt AI to generate a function that displays all  
Automorphic numbers between 1 and 1000 using a for loop.  
• Instructions:  
o Get AI-generated code to list Automorphic numbers using  
a for loop.  
o Analyze the correctness and efficiency of the generated  
logic.  
o Ask AI to regenerate using a while loop and compare both  
implementations

PROMPT :

write a python function to generate a function that displays all Automorphic numbers between 1 and 1000 using a for loop and regenerate using a while loop and compare both implementations

CODE AND OUTPUT :

A screenshot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer code

AI-generated content may be incorrect.

OBSERVATIONS :

* Both the for loop and while loop implementations successfully identified the Automorphic numbers between 1 and 1000 as 1, 5, 6, 25, 76, 376, and 625.
* For this specific task with a known number of iterations (1 to 1000), both for and while loops are suitable.
* The for loop implementation is more concise and generally considered more "Pythonic" for iterating over a fixed range or sequence as it handles initialization, condition checking, and incrementing implicitly.
* The while loop provides more explicit control over the loop's execution flow but requires manual management of the loop variable (initialization, condition, increment).

TASK 3 :

Ask AI to write nested if-elif-else conditions to classify  
online shopping feedback as Positive, Neutral, or Negative based  
on a numerical rating (1–5).  
• Instructions:  
o Generate initial code using nested if-elif-else.  
o Analyze correctness and readability.  
o Ask AI to rewrite using dictionary-based or match-case  
structure.

PROMPT :

write a python function of nested if-elif-else conditions to classify online shopping feedback as Positive, Neutral, or Negative based on a numerical rating and rewrite using dictionary-based or match-case structure.

CODE AND OUTPUT :

A screenshot of a computer

AI-generated content may be incorrect.

OBSERVATIONS :

* The classify\_feedback\_nested\_if function successfully classifies ratings: 5 as 'Positive', 3 as 'Neutral', 1 as 'Negative', and 'None' as 'Invalid RatinG
* The classify\_feedback\_dict\_match function accurately classifies valid integer ratings (1-5) and correctly identifies non-integer or out-of-range inputs as 'Invalid Rating'.
* After modifying the classify\_feedback\_nested\_if function to include input type validation, both functions produced identical classification results for a diverse set of test inputs.
* For this specific task, the dictionary-based approach is considered more readable and maintainable due to its clear mapping of ratings to classifications, compared to the sequential conditional checks of the if-elif-else structure.
* The performance difference between the two approaches is negligible for this task scale.

A screenshot of a computer

AI-generated content may be incorrect.

TASK 4 :

Generate a function using AI that displays all prime  
numbers within a user-specified range (e.g., 1 to 500).  
• Instructions:  
o Get AI-generated code to list all primes using a for loop.  
o Analyze the correctness and efficiency of the prime-  
checking logic.  
o Ask AI to regenerate an optimized version (e.g., using the  
square root method

PROMPT :

write a python function using loops that displays all prime numbers within a range 1 to 500 and regenerate using the square root method

CODE AND OUTPUT :

A screenshot of a computer program

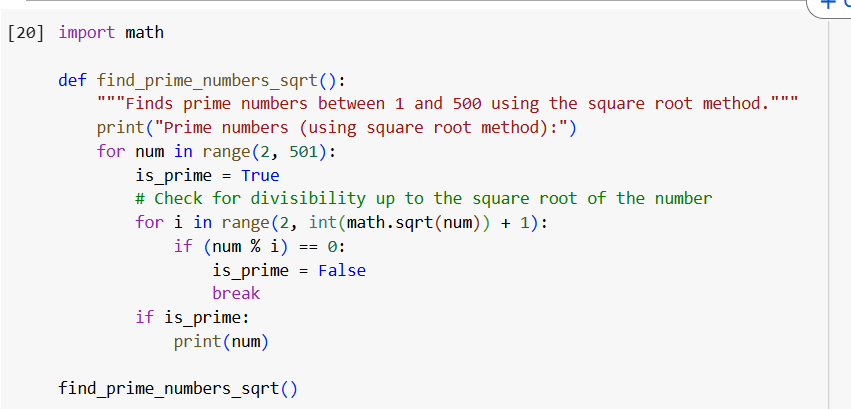
AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.





A screenshot of a computer

AI-generated content may be incorrect.

TASK 5 :

Use AI to build a Library class with methods to  
add\_book(), issue\_book(), and display\_books().  
• Instructions:  
o Generate Library class code using AI.  
o Analyze if methods handle edge cases (e.g., issuing  
unavailable books).  
o Ask AI to add comments and documentation

PROMPT :

write a python function to build a Library class with methods to add\_book(), issue\_book(), and display\_books() and add comments and documentation

CODE AND OUTPUT :

A screenshot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.