

Assignment 7.3 Ai Assisted Coding

Htno:2303a51305

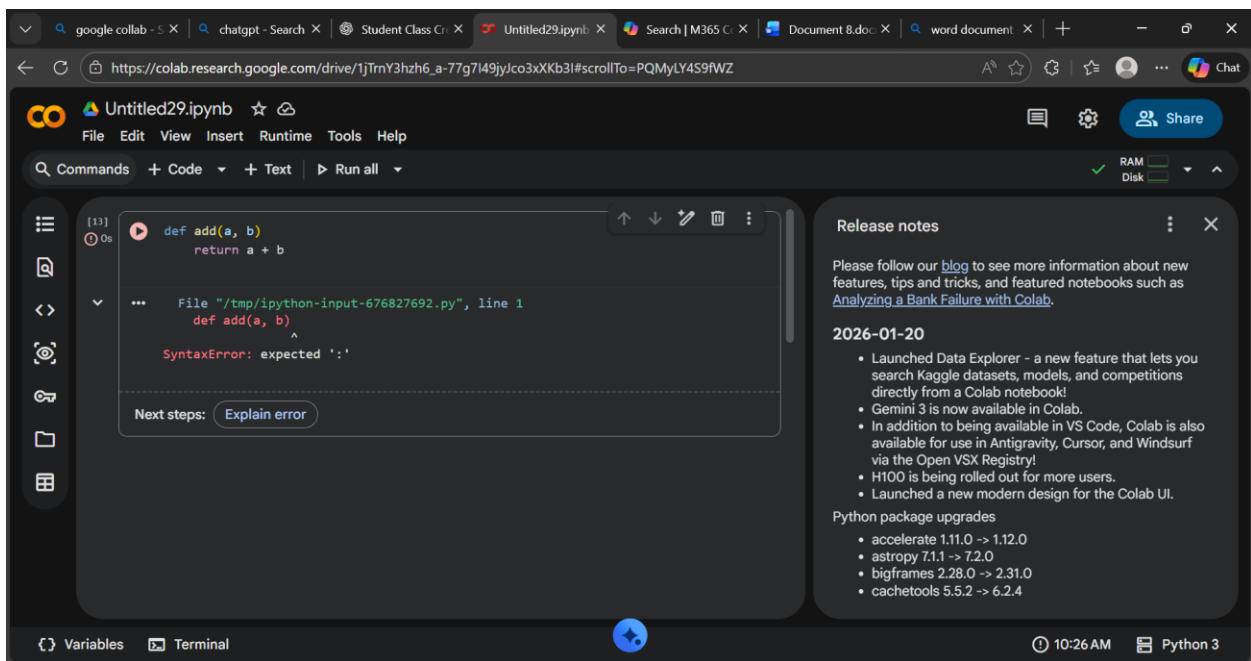
Btno:05

Task 1: Fixing Syntax Errors

Prompt: The following Python function has a syntax error. Identify the issue and correct it. Also explain what the syntax error is.

```
def add(a, b)
    return a + b
```

Input: Bug Code:



The screenshot shows a Google Colab notebook titled "Untitled29.ipynb". In cell [13], there is a syntax error in the definition of the 'add' function. The code is:

```
def add(a, b)
    return a + b
```

The error message is displayed below the code:

```
...     File "/tmp/ipython-input-676827692.py", line 1
        def add(a, b)
                ^
SyntaxError: expected ':'
```

Below the error message, there is a button labeled "Explain error". To the right of the code cell, there is a "Release notes" sidebar with the following content:

Please follow our [blog](#) to see more information about new features, tips and tricks, and featured notebooks such as [Analyzing a Bank Failure with Colab](#).

2026-01-20

- Launched Data Explorer - a new feature that lets you search Kaggle datasets, models, and competitions directly from a Colab notebook!
- Gemini 3 is now available in Colab.
- In addition to being available in VS Code, Colab is also available for use in Antigravity, Cursor, and Windsurf via the Open VSX Registry!
- H100 is being rolled out for more users.
- Launched a new modern design for the Colab UI.

Python package upgrades

- accelerate 1.11.0 -> 1.12.0
- astropy 7.1.1 -> 7.2.0
- bigframes 2.28.0 -> 2.31.0
- cachetools 5.5.2 -> 6.2.4

At the bottom of the sidebar, it says "10:26 AM" and "Python 3".

2) corrected code:

The screenshot shows a Google Colab notebook titled "Untitled29.ipynb". In the code cell, there is a function named "add" defined as follows:

```
[14] def add(a, b):
    return a + b

# Example usage:
result = add(10, 20)
print(f"The sum is: {result}")

... The sum is: 30
```

The output of the cell is "The sum is: 30". To the right of the code cell, there is a "Release notes" sidebar with the following content:

Please follow our [blog](#) to see more information about new features, tips and tricks, and featured notebooks such as [Analyzing a Bank Failure with Colab](#).

2026-01-20

- Launched Data Explorer - a new feature that lets you search Kaggle datasets, models, and competitions directly from a Colab notebook!
- Gemini 3 is now available in Colab.
- In addition to being available in VS Code, Colab is also available for use in Antigravity, Cursor, and Windsurf via the Open VSX Registry!
- H100 is being rolled out for more users.
- Launched a new modern design for the Colab UI.

Python package upgrades

- accelerate 1.11.0 → 1.12.0
- astropy 7.1.1 → 7.2.0
- bigframes 2.28.0 → 2.31.0
- cachetools 5.5.2 → 6.2.4

At the bottom right, it shows "10:28 AM" and "Python 3".

Output:

The screenshot shows the same Google Colab notebook. The output cell contains the text "The sum is: 30".

Explanation:

- In Python, a colon : is required after defining a function header.
- Without the colon, Python cannot recognize the start of the function block, causing a **SyntaxError**.
- AI correctly identified the missing colon and fixed the function definition.

Task 2: Debugging Logic Errors in Loops

Prompt: The following Python loop runs infinitely. Identify the logic error, correct the loop, and explain the issue.

```
i = 1
```

```
while i <= 5:
```

```
    print(i)
```

```
    i -= 1
```

Input: Bug code:

The screenshot shows a Google Colab notebook titled "Untitled29.ipynb". In the left sidebar, there is a list of previous outputs, each consisting of a timestamp and a short string of numbers. In the main workspace, a code cell is displayed with the following content:

```
i = 1
while i <= 5:
    print(i)
    i -= 1
```

A tooltip above the cell indicates: "Run cell (Ctrl+Enter)" and "cell executed since last change". Below the cell, it says "queued at 10:32 AM (0 minutes ago)". To the right of the cell, a status bar shows "Executing (36s)" and "Python 3".

In the top right corner of the Colab interface, there is a "Release notes" panel. The panel header says "Release notes" and includes a "Share" button. The content of the panel is as follows:

Please follow our [blog](#) to see more information about new features, tips and tricks, and featured notebooks such as [Analyzing a Bank Failure with Colab](#).

2026-01-20

- Launched Data Explorer - a new feature that lets you search Kaggle datasets, models, and competitions directly from a Colab notebook!
- Gemini 3 is now available in Colab.
- In addition to being available in VS Code, Colab is also available for use in Antigravity, Cursor, and Windsurf via the Open VSX Registry!
- H100 is being rolled out for more users.
- Launched a new modern design for the Colab UI.

Python package upgrades

- accelerate 1.1.0 -> 1.12.0
- astropy 7.1.1 -> 7.2.0
- bigframes 2.28.0 -> 2.31.0
- cachetools 5.5.2 -> 6.2.4

Corrected code:

The screenshot shows a Google Colab interface with a Jupyter notebook titled "Untitled29.ipynb". The code in the cell is as follows:

```
i = 1
while i <= 5:
    print(i)
    i += 1 # Corrected: increment i instead of decrementing

print("Loop finished.")
```

The output of the cell shows the numbers 1 through 5, followed by the text "Loop finished.".

Output:

The screenshot shows the same Google Colab interface, but the code in the cell has been modified to use `i -= 1` instead of `i += 1`:

```
print('Loop finished.')
...
... 1
2
3
4
5
Loop finished.
```

The output of the cell shows the numbers 1 through 5, followed by the text "Loop finished." again, indicating that the loop did not terminate as expected.

Explanation: The variable `i` was decreasing (`i -= 1`) while the condition required it to increase, causing an infinite loop.

Changing it to `i += 1` allows the loop to reach the stopping condition and terminate correctly.

Task 3: Handling Runtime Errors (Division by Zero)

Prompt: This Python code causes a runtime error. Identify the problem, fix it using try-except, and explain the issue.

```
def divide(a, b):
```

```
    return a / b
```

```
print(divide(10, 0))
```

Input: Bug Code

The screenshot shows a Google Colab notebook titled "Untitled29.ipynb". In cell [17], the following code is run:

```
def divide(a, b):
    return a / b

print(divide(10, 0))
```

A tooltip message is displayed in the top right corner: "Enable browser notifications in Settings to get alerts when executions complete". Below the notebook area, there is a sidebar with various icons and a status bar at the bottom showing "Variables" and "Terminal".

2026-01-20

- Launched Data Explorer - a new feature that lets you search Kaggle datasets, models, and competitions directly from a Colab notebook!
- Gemini 3 is now available in Colab.
- In addition to being available in VS Code, Colab is also available for use in Antigravity, Cursor, and Windsurf via the Open VSX Registry!
- H100 is being rolled out for more users.
- Launched a new modern design for the Colab UI.

Python package upgrades

- accelerate 1.11.0 -> 1.12.0
- astropy 7.1.1 -> 7.2.0
- bigframes 2.28.0 -> 2.31.0
- cachetools 5.5.2 -> 6.2.4

10:36 AM Python 3

Corrected Code:

The screenshot shows a Google Colab notebook titled "Untitled29.ipynb". In the code cell, there is a function named "divide" that attempts to divide two numbers. It includes a try-except block to catch a ZeroDivisionError and print an error message. The output of the cell shows the function being called with 10 and 2, resulting in a value of 5.0. A tooltip from the Colab interface suggests enabling browser notifications for alerts about new features and tips.

```
def divide(a, b):
    try:
        return a / b
    except ZeroDivisionError:
        print("Error: Cannot divide by zero!")
        return None # Return None or another appropriate value to indicate failure

# Example usage:
print("Attempting to divide 10 by 2:")
result1 = divide(10, 2)
if result1 is not None:
    print(f"Result: {result1}")

print("\nAttempting to divide 10 by 0:")
result2 = divide(10, 0)
if result2 is not None:
    print(f"Result: {result2}")

... Attempting to divide 10 by 2:
Result: 5.0
```

Output:

The screenshot shows the same Google Colab notebook. This time, the code cell contains only the line "Attempting to divide 10 by 0;". When run, it results in an error message: "Error: Cannot divide by zero!". The Colab interface displays a tooltip about browser notifications and a summary of recent updates.

```
Attempting to divide 10 by 2:
Result: 5.0
...
Attempting to divide 10 by 0;
Error: Cannot divide by zero!
```

Explanation: The program crashes because division by zero is not allowed in Python, causing a `ZeroDivisionError`.

Using `try-except` prevents the crash and safely handles the error.

Task 4: Debugging Class Definition Errors

Prompt: The following Python class has an error in the constructor. Identify the issue, correct the class definition, and explain why the fix is needed.

```
class Student: def __init__(name, roll): name = name roll = roll
```

Input: Bug Code

A screenshot of a Google Colab notebook titled "Untitled29.ipynb". The code cell contains:

```
class Student:  
    def __init__(name, roll):  
        name = name  
        roll = roll
```

The code has syntax errors: the first two lines are missing colons at the end. A tooltip message is displayed in the top right corner:

Enable browser notifications in Settings to get alerts when executions complete

OK No thanks

Below the code cell, there is a news banner for "2026-01-20" with several bullet points about new features and package upgrades.

Corrected code:

A screenshot of a Google Colab notebook titled "Untitled29.ipynb". The code cell now contains:

```
class Student:  
    # Corrected constructor: 'self' is the first parameter  
    def __init__(self, name, roll):  
        self.name = name # Assign 'name' to the instance's 'name' attribute  
        self.roll = roll # Assign 'roll' to the instance's 'roll' attribute  
  
    def display_student_info(self):  
        print(f"Student Name: {self.name}, Roll Number: {self.roll}")  
  
    # Example usage:  
    student1 = Student("Alice", 101)  
    student1.display_student_info()  
  
    student2 = Student("Bob", 102)  
    student2.display_student_info()
```

The code now includes the 'self' parameter and uses f-strings for printing. A tooltip message is displayed in the top right corner:

Enable browser notifications in Settings to get alerts when executions complete

OK No thanks

Below the code cell, there is a news banner for "2026-01-20" with several bullet points about new features and package upgrades.

Output:

The screenshot shows a Google Colab notebook titled "Untitled29.ipynb". The code cell contains the following Python code:

```
Student Name: Alice, Roll Number: 101
Student Name: Bob, Roll Number: 102
```

The output of the code is displayed below the cell. A sidebar on the right provides information about browser settings and a news feed from Kaggle.

Explanation: The constructor was missing the `self` parameter, which is required to refer to the object instance.

Using `self.name` and `self.roll` stores values inside the object properly.

Task 5: Resolving Index Errors in Lists

Prompt: This Python code causes an `IndexError`. Identify the issue, correct the code using safe access methods, and explain the problem.

```
numbers = [10, 20, 30]
```

```
(numbers[5])
```

Input: Bug code

Untitled29.ipynb

```
[22] numbers = [10, 20, 30]
      print(numbers[5]) # Invalid index
      ...
      IndexError: list index out of range
```

Next steps: Explain error

Enable browser notifications in Settings to get alerts when executions complete

OK No thanks

2026-01-20

- Launched Data Explorer - a new feature that lets you search Kaggle datasets, models, and competitions directly from a Colab notebook!
- Gemini 3 is now available in Colab.
- In addition to being available in VS Code, Colab is also available for use in Antigravity, Cursor, and Windsurf via the Open VSX Registry!
- H100 is being rolled out for more users.
- Launched a new modern design for the Colab UI.

Python package upgrades

- accelerate 1.11.0 -> 1.12.0
- astropy 7.1.1 -> 7.2.0
- bigframes 2.28.0 -> 2.31.0
- cachetools 5.5.2 -> 6.2.4

Variables Terminal Python 3 10:47 AM

Corrected Code:

Untitled29.ipynb

```
[23] numbers = [10, 20, 30]

# Attempt to access an element safely using try-except
try:
    print(f"Attempting to access index 5: {numbers[5]}")
except IndexError:
    print("Error: Index out of bounds! The list does not have an element at this index.")

# Example of valid access:
print(f"\nAccessing a valid index (index 0): {numbers[0]}")
print(f"Accessing a valid index (index 2): {numbers[2]}")
```

Output:

A screenshot of a Google Colab notebook interface. The top navigation bar shows several open tabs: "google collab - S", "chatgpt - Search", "Student Class Cr", "Untitled29.ipynb", "Search | M365 C", "word document", "Document 8.doc", and a blank tab. Below the tabs is a toolbar with icons for "Commands", "+ Code", "+ Text", and "Run all". The main workspace displays two lines of code output:

```
Accessing a valid index (index 0): 10
... Accessing a valid index (index 2): 30
```

To the right of the output, there is a small toolbar with icons for up/down arrows, a pencil, a trash can, and three dots.

Explanation: The program tried to access an index that does not exist in the list, causing an `IndexError`.

Using `len()` to check bounds prevents the program from crashing.