Guideline for student

Apply AI to learn deadlock

Step 1: Understand the Basics of Deadlock

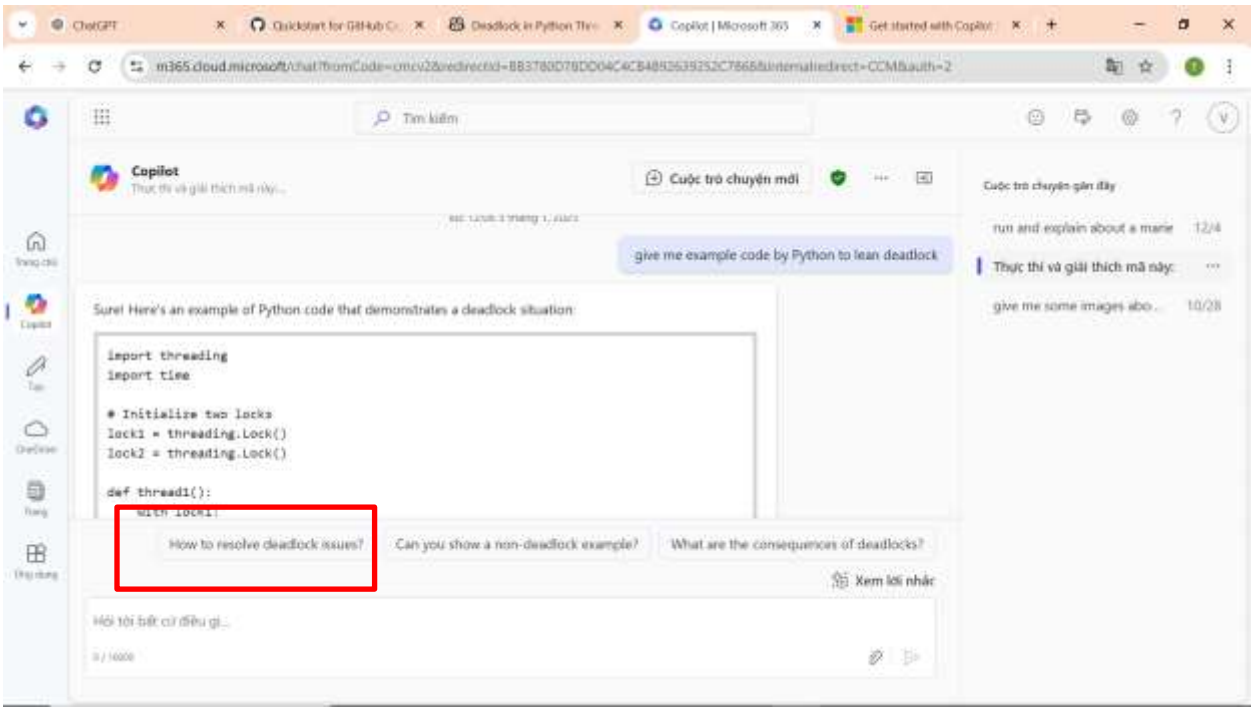
Before diving into AI-driven experimentation, make sure you understand the foundational concepts:

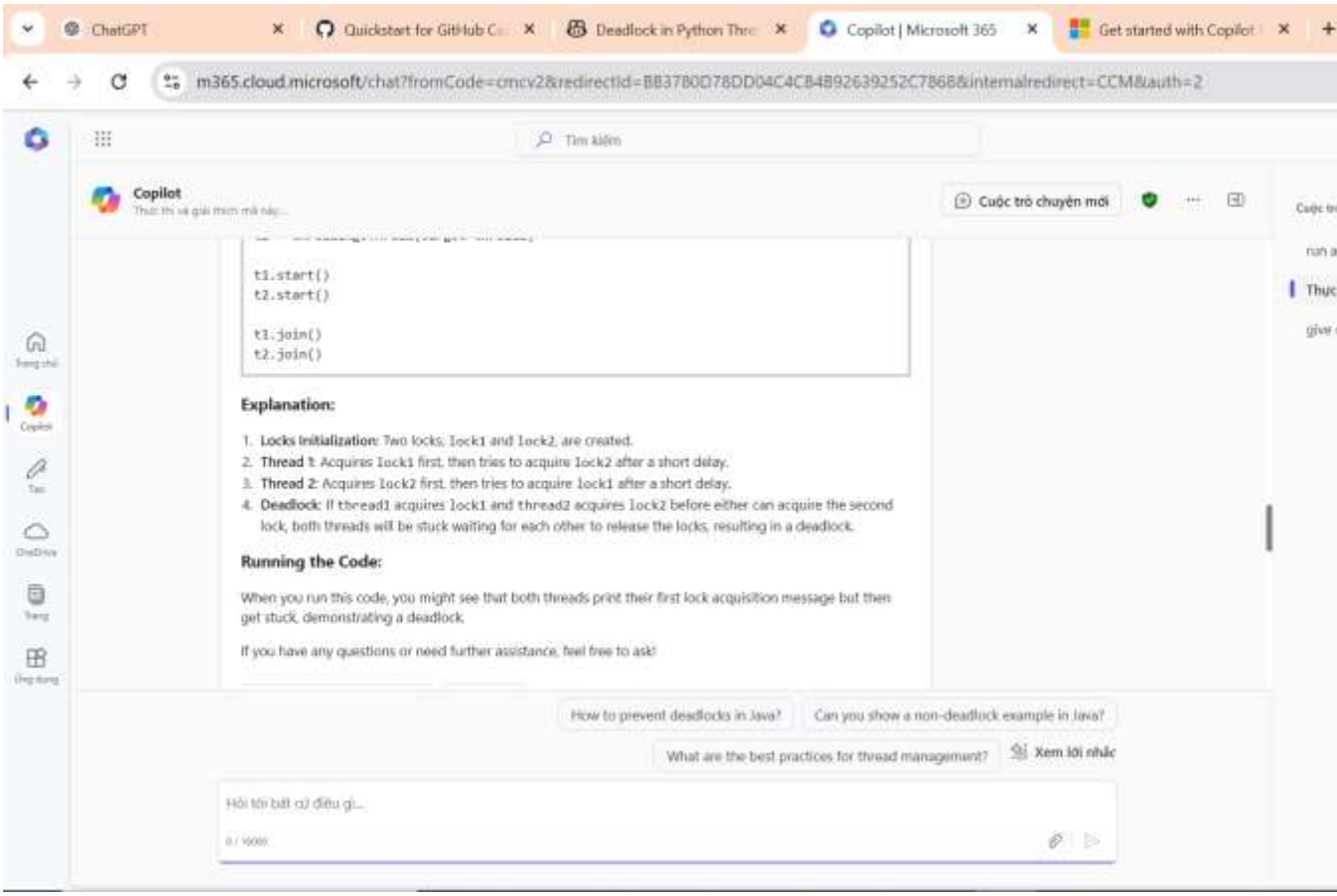
1. **Definition**: A deadlock occurs when a group of processes are waiting for resources held by each other, and none of them can proceed.
2. **Necessary Conditions** (Coffman conditions):
    - **Mutual Exclusion**: Resources are non-shareable.
    - **Hold and Wait**: A process holding a resource is waiting for additional resources.
    - **No Preemption**: Resources cannot be forcibly taken.
    - **Circular Wait**: A set of processes are waiting for each other in a circular chain.
3. **Deadlock Handling Strategies**:
    - **Prevention**: Alter system design to avoid one or more Coffman conditions.
    - **Avoidance**: Use algorithms like the Banker's Algorithm to avoid unsafe states.
    - **Detection and Recovery**: Detect deadlocks and terminate or preempt processes.
    - **Ignorance**: Simply ignore the problem (used in systems like UNIX).

Step 2: use AI Tools and Simulations with code (use Copilot/chatGPT)

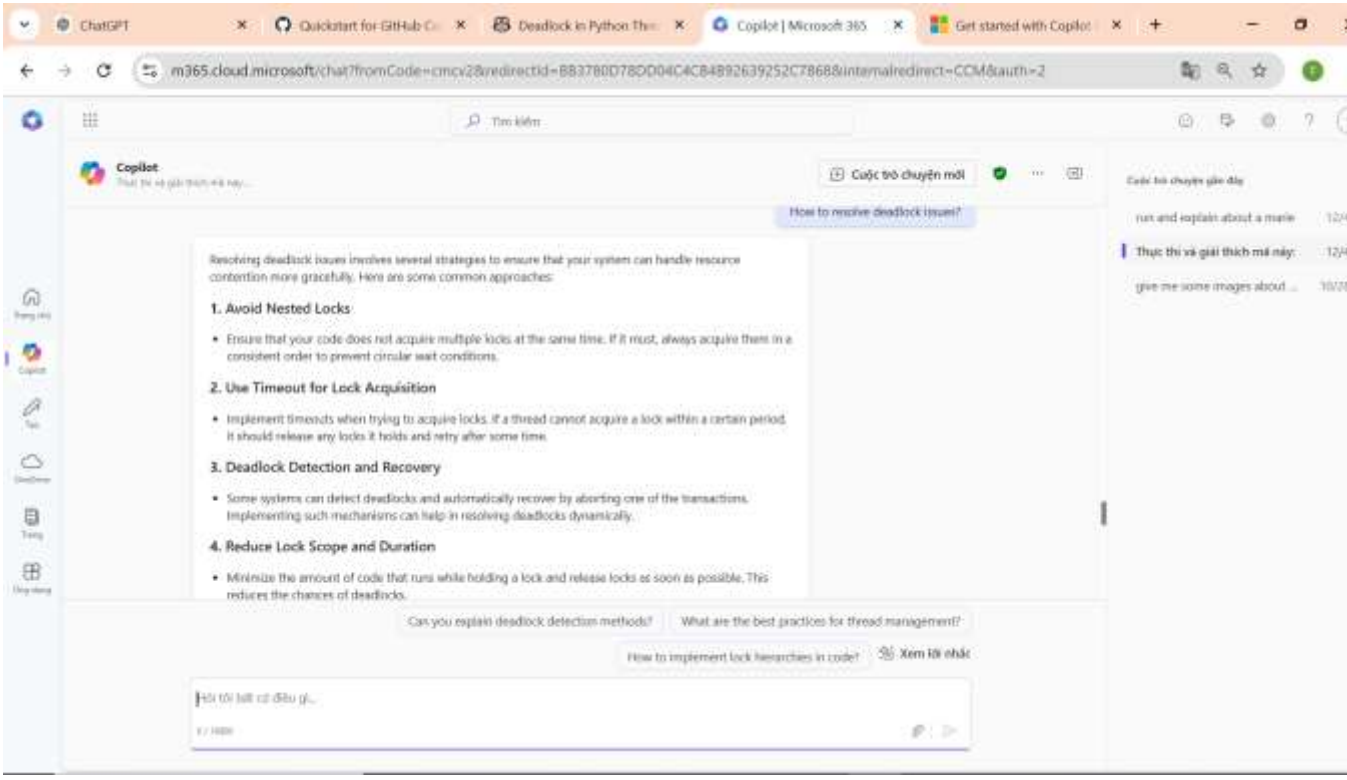
Requirements(use Copilot)

- Use Copilot with an account Microsoft (…@fe.edu.vn)
    - Open page: https://learn.microsoft.com/en-us/training/paths/copilot-power-platform/; and then click the button "Start"
    - At prompt: enter keyword "give me example code by Python to simulator deadlock"
      For example:

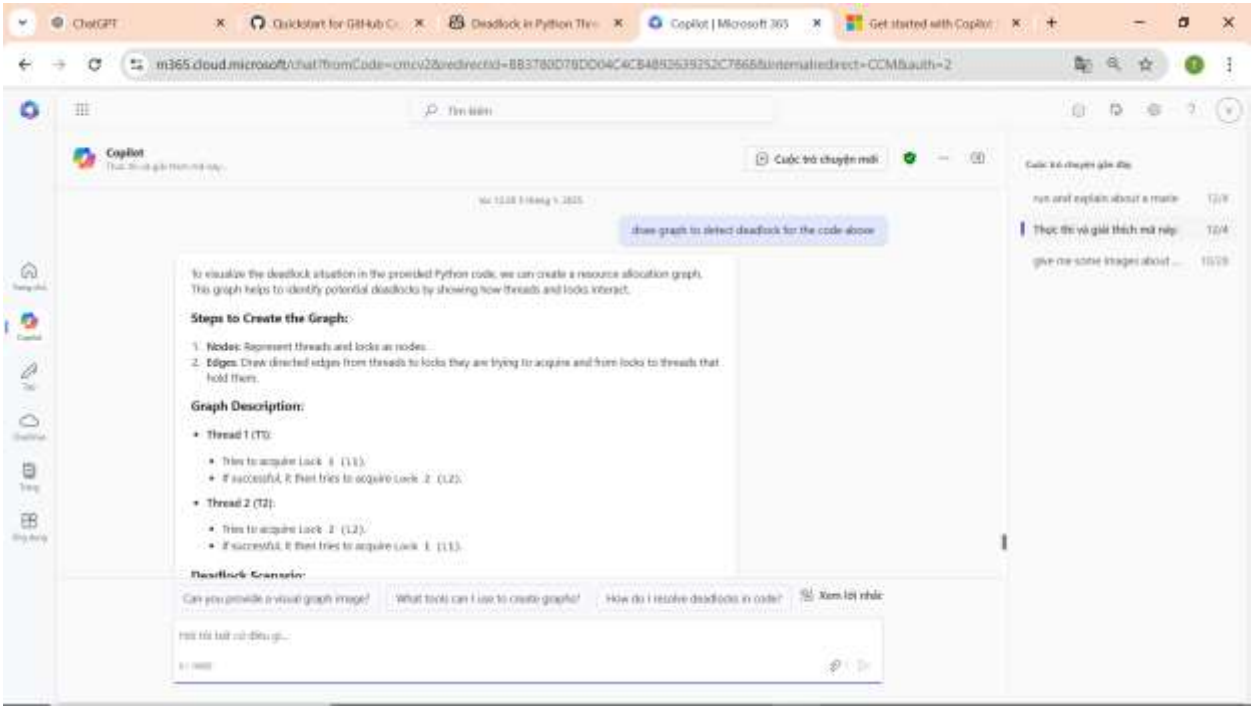

Copilot
Cuộc trò chuyện mới
give me example code by Python to lean deadlock
Sure! Here's an example of Python code that demonstrates a deadlock situation:
import threading
import time
# Initialize two locks
lock1 = threading.Lock()
lock2 = threading.Lock()
def thread1():
How to resolve deadlock issues?
Can you show a non-deadlock example?
What are the consequences of deadlocks?
Xem lời nhắc
Hỏi tôi bất cứ điều gì...


ChatGPT
Quickstart for GitHub C
Deadlock in Python Thre
Copilot | Microsoft 365
Get started with Copilot
m365.cloud.microsoft/chat?fromCode=cmcv2&redirectId=BB3780D78DD04C4CB4B92639252C7868&internalredirect=CCM&auth=2
Copilot
Cuộc trò chuyện mới
t1.start()
t2.start()
t1.join()
t2.join()
Explanation:
1. Locks Initialization: Two locks, lock1 and lock2, are created.
2. Thread 1: Acquires lock1 first, then tries to acquire lock2 after a short delay.
3. Thread 2: Acquires lock2 first, then tries to acquire lock1 after a short delay.
4. Deadlock: If thread1 acquires lock1 and thread2 acquires lock2 before either can acquire the second lock, both threads will be stuck waiting for each other to release the locks, resulting in a deadlock.
Running the Code:
When you run this code, you might see that both threads print their first lock acquisition message but then get stuck, demonstrating a deadlock.
If you have any questions or need further assistance, feel free to ask!
How to prevent deadlocks in Java?
Can you show a non-deadlock example in Java?
What are the best practices for thread management?
Xem lời nhắc
Hỏi tôi bất cứ điều gì...

- Clicks prompt “how to resolve deadlock issues”. Result:



-enter prompt: “draw graph to detect deadlock for the code above”



- Require student use paper and pen to redraw the above graph
- Ask Copilot/chatGPT to improve the above code to prevent deadlock
- Optional: student can run code by Jupiter or any suitable environment

Step 3: Practical Deadlock Exercises

Exercise 1: Simulate a Deadlock

- Use Copilot/chatGPT to generate code to simulate a deadlock, analyze it, and then implement one or more of the following:
    - Deadlock Prevention (e.g., locking in a specific order).
    - Deadlock Avoidance (e.g., Banker's Algorithm).
    - Deadlock Detection (e.g., graph traversal to find cycles).

Exercise 2: Visualize Resource Allocation Graphs

- Use AI to help generate or visualize resource allocation graphs, showing processes and the resources they hold or request.

Exercise 3: Solve a Deadlock Scenario

- Describe a deadlock problem to AI (e.g., "Three processes P1, P2, P3 compete for two resources R1 and R2"). Let AI suggest possible resolutions or guide you in writing a program to simulate and solve the issue.