



## CSE470: Software Engineering

**SET - A**

Semester: **Summer 2025**  
Examination: **Make Up Quiz**

Time: **25 minutes**  
Full marks: **15**

Name: \_\_\_\_\_

ID: \_\_\_\_\_

Section: **8**

Read the code and answer the following questions:

### Python

```
def cooling_system_simulation():
    temperature = 50
    max_temperature = 80
    fan_speed = 5
    cooling_rate = 2
    is_cooling_system_on = False
    is_emergency_mode = False

    while temperature > 0:
        if is_cooling_system_on:
            for i in range(0, fan_speed, 1):
                if temperature < max_temperature:
                    for j in range(0, cooling_rate, 1):
                        temperature -= 1
                        print(f"Cooling... Current temperature: {temperature}")
                    else:
                        print("Temperature is too high!")
            else:
                is_cooling_system_on = False
                temperature += 1
                return f"Warming... Current temperature: {temperature}"

        if is_emergency_mode:
            print("Emergency mode activated! Forcing rapid cooling.")
            temperature -= 5

        if temperature <= 0:
            print("System shutting down. Temperature is safe.")
            break
```

#### A. Answer the following questions for the given method:

1. Mark the nodes on the question paper and draw the CFG [1+6]
2. Find out all the independent paths in your CFG [4]
3. Calculate the Cyclomatic Complexity [3]

#### B. What box is the path based testing? [1]

**White Box Testing**

— — — — —





## CSE470: Software Engineering

**SET - B**

Semester: **Summer 2025**  
Examination: **Make Up Quiz**

Time: **25 minutes**  
Full marks: **15**

Name: \_\_\_\_\_

ID: \_\_\_\_\_

Section: **8**

Imagine you are making a local video streaming service like youtube. When you created the LocalVideoPlayer, it could not stream video as efficiently as Youtube does. Mainly because you have used localDataStream() in your system and Youtube is using universalDataStream(). Now you want to make an updated platform BetterLocalVideoPlayer that follows the youtube data streaming standards while implementing the previous methods. You also need to make sure people cannot copy your service just like you copied Youtube. Then you want users to be able to signup in the newly created player. Once they have completed the signup, they can subscribe to the channel. For simplicity you can assume there will only be one channel in this player. The driver code is given below for your reference.

Driver Code	Output
<pre> newPlayer = BetterLocalVideoPlayer() print("====1====") user1 = User("User 1") user2 = User("User 2") print("====2====") user1.subscribe() print("====3====") user1.signup() print("====4====") user1.subscribe() print("====5====") user2.signup() print("====6====") newPlayer.uploadNewVideo() print("====7====") newPlayer.universalDataStream() print("====8====") user1.deleteAccount() print("====9====") newPlayer.uploadNewVideo() print("====10====") user2.subscribe() print("====11====") newPlayer.uploadNewVideo() print("====12====") user2.unsubscribe() print("====13====") newPlayer.uploadNewVideo() </pre>	<pre> =====1===== =====2===== User did not signup =====3===== =====4===== User 1 has subscribed to Channel A =====5===== =====6===== Notification (User 1): New Video Uploaded to Channel A =====7===== Streaming Video =====8===== =====9===== =====10===== User 2 has subscribed to Channel A =====11===== Notification (User 2): New Video Uploaded to Channel A =====12===== User 2 has unsubscribed to Channel A =====13===== =====14===== </pre>

1. What design pattern(s) should be used in this scenario? [3] **Singleton+Observer+Adapter**
2. Implement the necessary classes in such a way that the driver code generates the given output. [12]

**Answer notebook:** [Design Patterns Quiz Solutions](#)