```
# Subject
class StockPulse:
  def __init__(self):
    self.followers={}

  def set_stock_price(self, company, price):
    self.notify_subscribers(company, price)

  def subscribe(self, subscriber, company):
    if company not in self.followers.keys():
      self.followers[company] = [subscriber]
    else:
      self.followers[company].append(subscriber)

  def unsubscribe(self, subscriber, company):
    print(f"{subscriber.name} unsubscribed from {company}")
    self.followers[company].remove(subscriber)

  def notify_subscribers(self, company, prices):
    for subscriber in self.followers[company]:
      subscriber.notify(company, prices)

# Observer
class Trader:
  def __init__(self, name):
    self.name = name

  def notify(self, company, stockPrice):
    print(f"{self.name} received update on stock price: {company} - {stockPrice}")

# Centralized stock system
stock_pulse = StockPulse()

# Create traders
alice = Trader("Alice")
bob = Trader("Bob")

# Subscribe traders to stocks
stock_pulse.subscribe(alice, "AAPL")  # Alice follows Apple
stock_pulse.subscribe(bob, "GOOG")    # Bob follows Google
stock_pulse.subscribe(alice, "GOOG")  # Alice also follows Google

# Update stock prices
print("1==================")
stock_pulse.set_stock_price("AAPL", 150.0)  # Notifies Alice
print("2==================")
stock_pulse.set_stock_price("GOOG", 2800.0) # Notifies both Alice and Bob

# Bob unsubscribes from Google stock
print("3==================")
stock_pulse.unsubscribe(bob, "GOOG")

# Update stock prices again
print("4==================")
stock_pulse.set_stock_price("GOOG", 2850.0) # Notifies only Alice
```

```
1==================
Alice received update on stock price: AAPL - 150.0
2==================
Bob received update on stock price: GOOG - 2800.0
Alice received update on stock price: GOOG - 2800.0
3==================
Bob unsubscribed from GOOG
4==================
Alice received update on stock price: GOOG - 2850.0
```

```
class A:
  def methodA(self):
    print("Instance Method A")


  @staticmethod
  def methodC():
    print("Static Method C")
a = A()

A.methodC()
```

```
Static Method C
```