

```
# Assuming bid_price and ask_price are already defined
bid_price = 100.0 # replace with the actual bid price
ask_price = 110.0 # replace with the actual ask price

# Compute the new price using the formula: (bid_price + ask_price) / 2
price = (bid_price + ask_price) / 2

# Print the result
print("The computed price is:", price)
```

The computed price is: 105.0

```
def compute_stock_ratio(stock_price_a, stock_price_b):
    # Check if stock_price_b is 0 to avoid division by zero
    if stock_price_b == 0:
        print("Error: Cannot divide by zero.")
        return None # You can handle this case according to your
requirements
```

```
    # Compute the ratio of stock_price_a to stock_price_b
    ratio = stock_price_a / stock_price_b
```

```
    # Return the computed ratio
    return ratio
```

```
# Example usage:
```

```
stock_price_a = 120.0 # replace with the actual stock price for stock
A
```

```
stock_price_b = 100.0 # replace with the actual stock price for stock
B
```

```
# Call the function and print the result
```

```
result = compute_stock_ratio(stock_price_a, stock_price_b)
```

```
if result is not None:
```

```
    print("The computed stock ratio is:", result)
```

The computed stock ratio is: 1.2

```
def getDataPoint(stock_name):
```

```
    # This is a placeholder function, replace it with your actual
implementation
```

```
    # It should return the current price for the given stock_name
```

```
    # For example, you might get the price from an API or database
```

```
    # For simplicity, let's use random values as placeholders
```

```
    import random
```

```
    return random.uniform(100, 200)
```

```
def compute_stock_ratio(stock_price_a, stock_price_b):
```

```
    if stock_price_b == 0:
```

```
        print("Error: Cannot divide by zero.")
```

```
        return None
```

```

    return stock_price_a / stock_price_b

# Main method
def main():
    # List of stock names
    stock_names = ["StockA", "StockB", "StockC"]

    # Dictionary to store stock prices
    prices = {}

    # Iterate through each stock and get data points
    for stock_name in stock_names:
        # Get the current data point for the stock
        current_price = getDataPoint(stock_name)

        # Store the data point in the prices dictionary
        prices[stock_name] = current_price

    # Print the stored data points
    for stock_name, current_price in prices.items():
        print(f"{stock_name}: {current_price}")

    # Calculate and print ratios
    for stock_name_a in stock_names:
        for stock_name_b in stock_names:
            if stock_name_a != stock_name_b:
                ratio = compute_stock_ratio(prices[stock_name_a],
prices[stock_name_b])
                print(f"Ratio between {stock_name_a} and
{stock_name_b}: {ratio}")

# Run the main method
if __name__ == "__main__":
    main()

```

```

StockA: 147.84118959894488
StockB: 171.85186895138418
StockC: 117.11097621236745
Ratio between StockA and StockB: 0.8602826987047096
Ratio between StockA and StockC: 1.262402503851148
Ratio between StockB and StockA: 1.1624085913917095
Ratio between StockB and StockC: 1.46742751627098
Ratio between StockC and StockA: 0.7921403807021534
Ratio between StockC and StockB: 0.6814646644634247

```