```
import java.util.ArrayList;
        import java.util.HashMap;
        import java.util.Map;
        import java.util.Scanner;

class Stock {
    private String symbol;
    private String name;
    private double currentPrice;
    private double change;

    public Stock(String symbol, String name, double currentPrice, double change) {
        this.symbol = symbol;
        this.name = name;
        this.currentPrice = currentPrice;
        this.change = change;
    }

    public String getSymbol() {
        return symbol;
    }

    public String getName() {
        return name;
    }

    public double getCurrentPrice() {
        return currentPrice;
    }

    public double getChange() {
        return change;
    }

    public void updatePrice(double newPrice) {
        change = (newPrice - currentPrice) / currentPrice * 100;
        currentPrice = newPrice;
    }

    @Override
    public String toString() {
        return symbol + ": " + name + " -> $" + currentPrice + " (" + change +
"%)";
    }
}

class Portfolio {
    private Map<String, Integer> stocks;
    private double cash;
```

```
    public Portfolio() {
        stocks = new HashMap<>();
        cash = 10000.0;
    }

    public void buyStock(Stock stock, int quantity) {
        double totalCost = stock.getCurrentPrice() * quantity;
        if (cash >= totalCost) {
            cash -= totalCost;
            stocks.put(stock.getSymbol(), stocks.getOrDefault(stock.getSymbol(), 0)
+ quantity);
            System.out.println("\n\tYou bought " + quantity + " shares of " +
stock.getName() + " at $" + stock.getCurrentPrice() + ".");
        } else {
            System.out.println("\n\tInsufficient cash balance!");
        }
    }

    public void sellStock(Stock stock, int quantity) {
        if (stocks.containsKey(stock.getSymbol()) && stocks.get(stock.getSymbol())
>= quantity) {
            cash += stock.getCurrentPrice() * quantity;
            stocks.put(stock.getSymbol(), stocks.get(stock.getSymbol()) -
quantity);
            if (stocks.get(stock.getSymbol()) == 0) {
                stocks.remove(stock.getSymbol());
            }
            System.out.println("\n\tYou sold " + quantity + " shares of " +
stock.getName() + " at $" + stock.getCurrentPrice() + ".");
        } else {
            System.out.println("\n\tNot enough shares to sell!");
        }
    }

    public void printPortfolio(ArrayList<Stock> allStocks) {
        System.out.println("\tPortfolio");
        System.out.println("\t---------");
        for (Map.Entry<String, Integer> entry : stocks.entrySet()) {
            String symbol = entry.getKey();
            int quantity = entry.getValue();
            Stock stock = getStockBySymbol(symbol, allStocks);
            if (stock != null) {
                System.out.println("\t" + stock.getName() + " (" + symbol + "): " +
quantity + " shares at $" + stock.getCurrentPrice() + " each.");
            }
        }
        System.out.println("\tCash: $" + cash);
    }
```

```
    public double getPortfolioValue(ArrayList<Stock> allStocks) {
        double value = cash;
        for (Map.Entry<String, Integer> entry : stocks.entrySet()) {
            Stock stock = getStockBySymbol(entry.getKey(), allStocks);
            if (stock != null) {
                value += stock.getCurrentPrice() * entry.getValue();
            }
        }
        return value;
    }

    private Stock getStockBySymbol(String symbol, ArrayList<Stock> allStocks) {
        for (Stock stock : allStocks) {
            if (stock.getSymbol().equals(symbol)) {
                return stock;
            }
        }
        return null;
    }
}

public class StockTradingPlatform {
    private ArrayList<Stock> stocks;
    private Portfolio portfolio;

    public StockTradingPlatform() {
        stocks = new ArrayList<>();
        portfolio = new Portfolio();

        stocks.add(new Stock("AAPL", "Apple Inc.", 150.0, 2.5));
        stocks.add(new Stock("GOOG", "Alphabet Inc.", 2500.0, 1.2));
        stocks.add(new Stock("MSFT", "Microsoft Corporation", 200.0, 3.1));
    }

    public void printMarketData() {
        for (Stock stock : stocks) {
            System.out.println("\t\t" + stock);
        }
    }

    public Stock getStockBySymbol(String symbol) {
        for (Stock stock : stocks) {
            if (stock.getSymbol().equals(symbol)) {
                return stock;
            }
        }
        return null;
    }

    public void updateMarketData() {
```

```
        for (Stock stock : stocks) {
            double newPrice = stock.getCurrentPrice() + (Math.random() - 0.5) * 10;
            stock.updatePrice(newPrice);
        }
    }

    public static void main(String[] args) {
        StockTradingPlatform platform = new StockTradingPlatform();
        Scanner scanner = new Scanner(System.in);

        while (true) {

System.out.println("\t+---------------------------------------------------------------
--------+");
            System.out.println("\t|\t\t\tSTOCK TRADING PLATFORM\t\t\t     |");

System.out.println("\t+---------------------------------------------------------------
--------+");

            System.out.println("\t\t[1] View Market Data\t\t[2] Buy Stock\n\t\t[3]
Sell Stock\t\t\t[4] Track Portfolio\n\t\t[5] Update Market Data\t\t[6] Exit");

            System.out.print("\n\tChoose an option: ");
            int option = scanner.nextInt();

            System.out.println("\n");

            switch (option) {
                case 1:

System.out.println("\t+---------------------------------------------------------------
--------+");
                    System.out.println("\t|\t\t\t      MARKET DATA\t\t\t     |");

System.out.println("\t+---------------------------------------------------------------
--------+");

                    platform.printMarketData();
                    System.out.println("\n");
                    break;
                case 2:

System.out.println("\t+---------------------------------------------------------------
--------+");
                    System.out.println("\t|\t\t\t       BUY STOCK\t\t\t     |");

System.out.println("\t+---------------------------------------------------------------
--------+");

                    System.out.print("\t\tEnter stock symbol: ");
```

```
                    String symbol = scanner.next();

                    Stock stock = platform.getStockBySymbol(symbol);
                    if (stock != null) {
                        System.out.print("\t\tEnter quantity: ");
                        int quantity = scanner.nextInt();
                        platform.portfolio.buyStock(stock, quantity);
                    } else {
                        System.out.println("\n\tStock not found!");
                    }
                    System.out.println("\n");
                    break;
                case 3:

System.out.println("\t+---------------------------------------------------------------
--------+");
                    System.out.println("\t|\t\t\t       SELL STOCK\t\t\t     |");

System.out.println("\t+---------------------------------------------------------------
--------+");

                    System.out.print("\t\tEnter stock symbol: ");
                    symbol = scanner.next();

                    stock = platform.getStockBySymbol(symbol);
                    if (stock != null) {
                        System.out.print("\t\tEnter quantity: ");
                        int quantity = scanner.nextInt();
                        platform.portfolio.sellStock(stock, quantity);
                    } else {
                        System.out.println("\n\tStock not found!");
                    }
                    System.out.println("\n");
                    break;
                case 4:
                    platform.portfolio.printPortfolio(platform.stocks);
                    System.out.println("\tPortfolio Value: $" +
platform.portfolio.getPortfolioValue(platform.stocks) + "\n\n");
                    break;
                case 5:
                    platform.updateMarketData();
                    System.out.println("\tMarket data updated.\n\n");
                    break;
                case 6:
                    System.out.println("\tExiting...");
                    scanner.close();
                    return;
                default:
                    System.out.println("\tInvalid option! Please try again.\n\n");
            }
```

```
            }
        }
    }
```