

Assignment 1

Stock-Order Matching Problem

Problem: To develop a software for matching stock orders and performing trades. Here matching is defined as matching “Buy” orders to “Sell” orders and vice versa.

Algorithm: We maintain a list of Orders called “Pending List” which maintains the orders not matched so far. The list is sorted on First Come First Serve basis i.e. ordered by time of the order. An order is matched starting from the beginning of the list according to parameters such as Order Id, Trade Type, Stock Name, Price, Quantity. For an order to be matched the following conditions must be followed:

- Order ID must be different for the two orders
- Trade Type should be different
- Stock Name must be the same
- (Buy Order) Price of the order matched in the pending list must be less or equal to the price of the incoming buy order.
- (Sell order) Price of the order matched in the pending list must be greater or equal to the price of the incoming sell order.

Design: The design of the software is described below.

We have 2 classes defined namely:

- Stock Market Class
- Order Class

These are to give the abstractions of an order and a stock market.

The stock market has a central pending list (In this case only). So our class has a data member representing this list.

The market receives the orders of buy and sell and matches them. For this we have 2 methods

- AddOrder - which adds an order to the market and matches it if possible.
- View List - This allows us to view the current state of the pending list

An order is represented by a six-tuple

<TimeStamp, CustomerID, Stock Name, Trade Type, Price, Quantity>

This is represented by a class named Order which has the above six data members.

This class has various functions for getting and setting these data members.

On a higher level, the software has been logically divided into 2 parts, functions of a stock market which are implemented by the member functions: AddOrder and View List and definition of an order which is done by the Order class.