Object Oriented Programming Coursework for Midterm: MerklerexBot

Introduction

In the course, we developed the Merklerex exchange simulation and a command-line user interface which allows a human to carry out trading activities such as bidding for and offering trades, market analysis and so on.

In this assignment you are tasked with creating a bot which can automatically carry out trading activities on the Merklerex platform. Essentially the bot takes the place of the human, interacting with the exchange to do the things that you were previously doing manually using the options on the menu.

Requirements

The purpose of the bot is to automate trading. The bot should start with a certain amount of currency and it should trade automatically with an aim to increasing the amount of currency it owns. The functionality of the bot should meet the following requirements:

- R1: Market analysis:

  - R1A: Bot accesses order book

  - R1B: Bot can generate predictions of future market prices

- R2: Bids, asks and sales

  - R2A: Bot can generate bids and place them in the order book

  - R2B: Bot can generate bids and place them in the order book

  - R2C: Bot updates wallet after successful sales

- R3: Logging

  - R3A: Bot generates a log of the contents of its wallet

  - R3B: Bot generates a log of its bids and asks

  - R3C: Bot generates a log of its sales

Code style and technique

Your code should be written according to the following style and technique guidelines:

- C1: Code is organised into header (.h or .hpp) files and implementation files (.cpp). Header files contain class interface definitions, cpp files contain implementations of class function memenbers.

- C2: Class interfaces in header files have comments for each public function describing purpose, inputs and outputs
- C3: Code is laid out clearly with consistent indenting
- C4: Functions, classes and variables have meaningful names, with a consistent naming style

Documentation

You should write a report and submit your source code. The submission should contain the following items and information:

- D1: Source code in standard ZIP format
- D2: Report in PDF format/ In your report, put the following:
  - For each requirement-sub requirement state how this was achieved or if it was not achieved. Explain where it can be found in the code. Use focused, short code extracts if they make your explanation clearer.
  - Algorithms: where you created algorithms, e.g. for predicting future market, explain each algorithm and state where it can be found in the code. Use focused, short code extracts if they make your explanation clearer.

Marking criteria

We will mark your work according to the set of criteria shown below, which consider the requirements, your programming technique and style and the documentation you have provided:

Category	Criterion	Not addressed	Attempted but did not fully meet requirements (if relevant)	Met requirement fully	Weight
Code style and technique	C1: Code is organised into header (.h or .hpp) files and implementation files (.cpp). Header files contain class interface definitions, cpp files contain implementations of class function memenbers.				
Code style and technique	C2: Class interfaces in header files have comments for each public function describing purpose, inputs and outputs				
Code style and technique	C3: Code is laid out clearly with consistent indenting				
Code style	C4: Functions, classes				

and technique	and variables have meaningful names, with a consistent naming style				
R1: Market analysis	R1A: Bot accesses order book				
	R1B: Bot can generate predictions of future market prices				
R2: Bids, asks and sales	R2A: Bot can generate bids and asks and place them in the order book				
	R2B: Bot updates wallet after successful sales				
R3: Logging	R3A: Bot generates a log of the contents of its wallet				
	R3B: Bot generates a log of its bids and asks				
	R3C: Bot generates a log of its sales				
Documentation	D1: source code as zip				
	D2: report as PDF				
	D3: reporting on requirements				
	D4: description of algorithms				