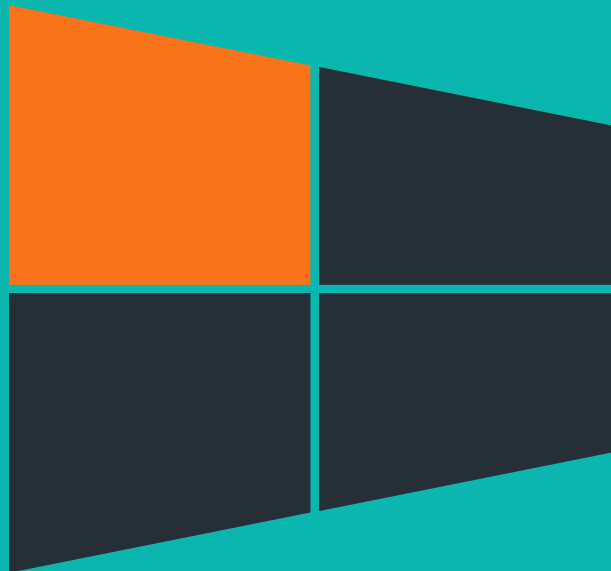


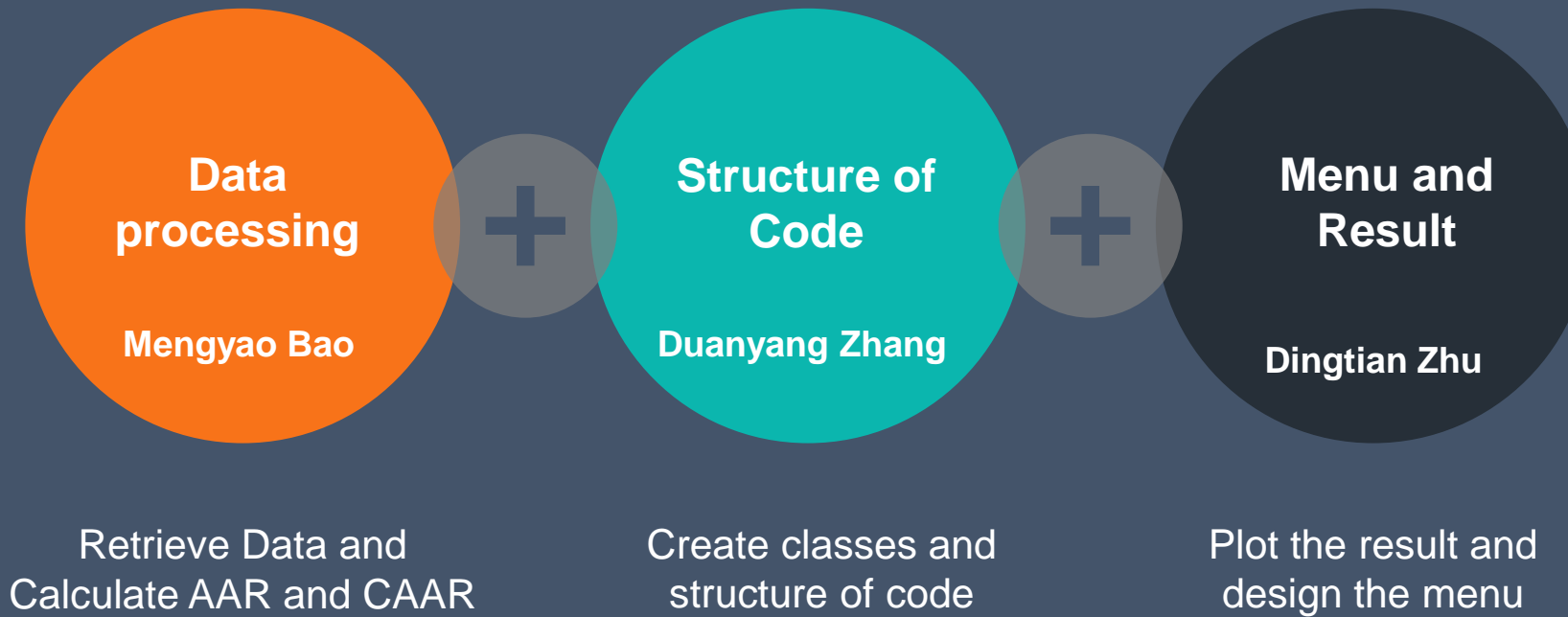
FRE 6883 Financial Computing Team Project

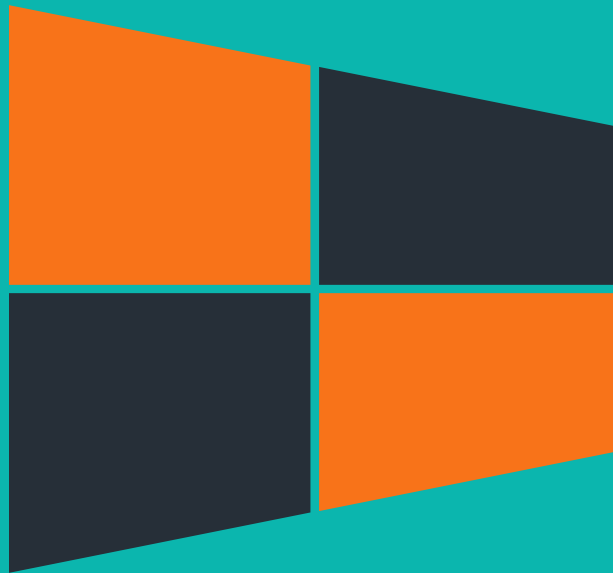
Mengyao Bao
Dingtian Zhu
Duanyang Zhang



Introduction

Structure





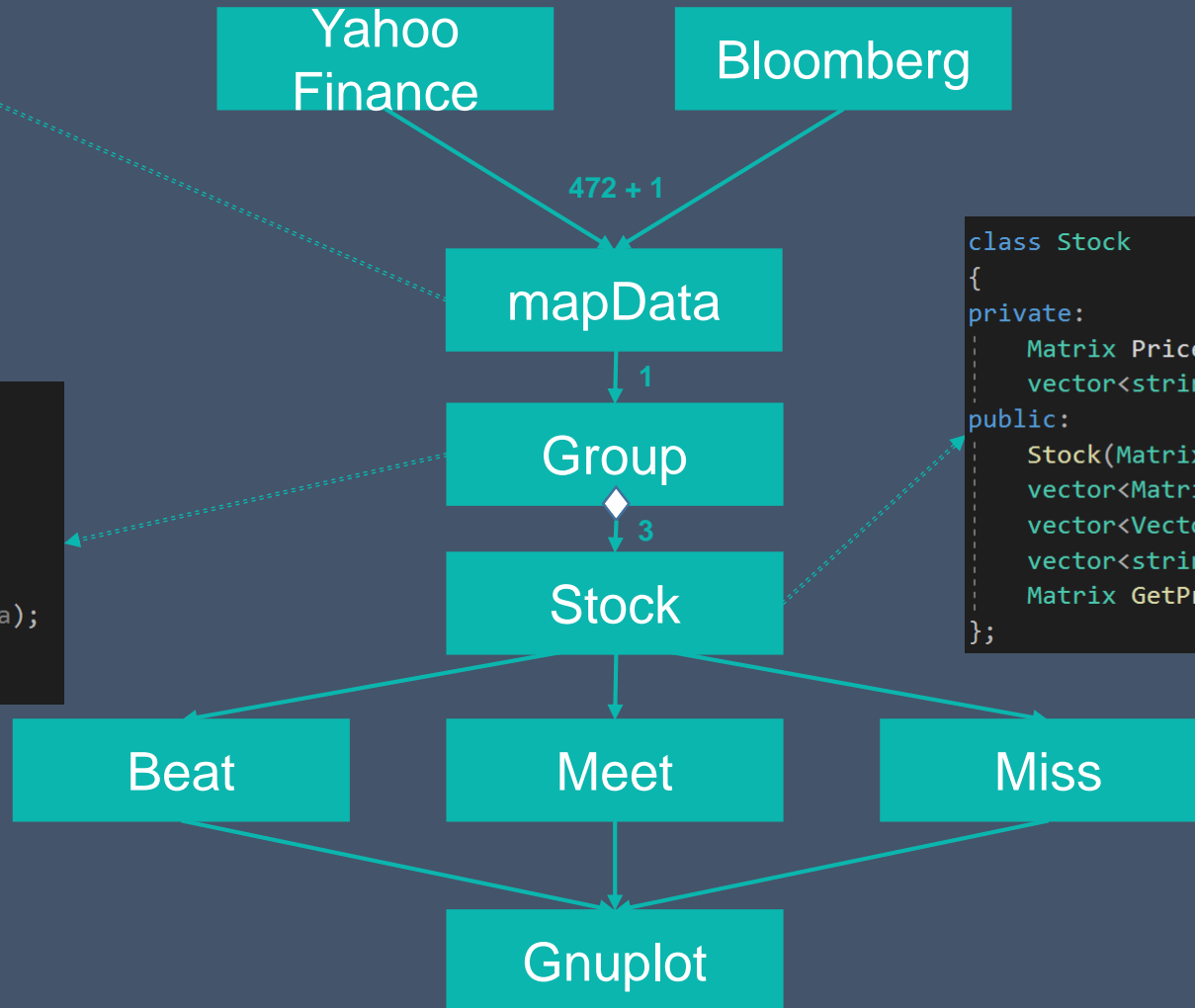
Design

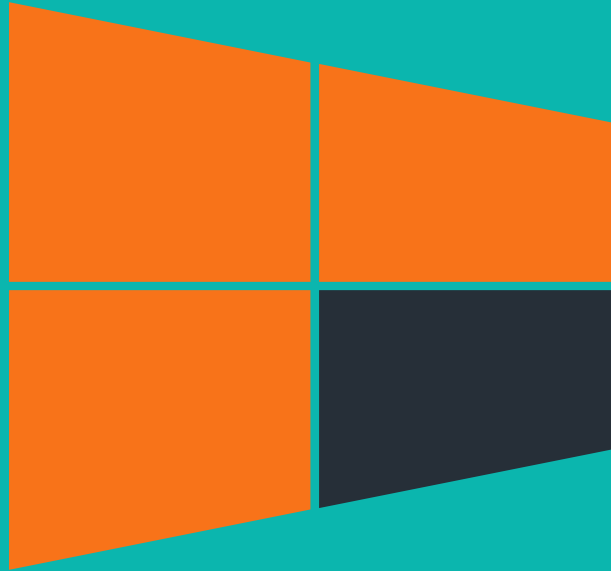
Structure

```
map<string, StockData> mapData;  
  
typedef struct STOCKDATA  
{  
    vector<string> sDate;  
    vector<float> sValue;  
    vector<float> sReturn;  
    vector<float> sAR;  
    string reportDate;  
    int reportIndex;  
    float sur;  
    int group;  
    string group_;  
}StockData;
```

```
class Group  
{  
public:  
    Stock* Beat;  
    Stock* Meet;  
    Stock* Miss;  
    Group(map<string, StockData>& mapdata);  
    ~Group();  
};
```

```
class Stock  
{  
private:  
    Matrix PriceMatrix;  
    vector<string> NameVector;  
public:  
    Stock(Matrix PriceMatrix_, vector<string> NameVector_);  
    vector<Matrix> BootStrap(int window);  
    vector<Vector> CalStd(int window);  
    vector<string> GetNameVector();  
    Matrix GetPriceMatrix();  
};
```





IMPLEMENTATION

Data

◆ Bloomberg Data:

- Saved in a csv file: stock index, quarter report date, earning surprise
- 472 stocks:
 - Components of SP500
 - Companies published report in the first quarter, 2019 (474 stocks from Bloomberg)
 - Delete NLOK and BKR, whose trading data have errors on Yahoo Finance

◆ Yahoo Finance:

- Historical trading data for stocks around quarter report date
[report date – 60 days, report date + 60days]
- Historical trading data of SPY
[end of Oct, 2018, end of May. 2019]

Data Retrieve

mapData:

Key: index (string); Value: StockData (struct)

DATA TYPE	DATA MEMBER
VECTOR<STRING>	sDate
VECTOR<FLOAT>	sValue
	sReturn
	sAR
STRING	reportDate
	group_
INT	reportIndex
	group
FLOAT	sur

Step1

Bloomberg Data

- Add 472+1 pairs into the map
- Each pair: [index, StockData]
- Only data member reportDate and sur are filled

Step2

Yahoo Finance

- Crawl a list of trading dates and adjusted prices
- Fill data member: sDate, sValue

Step3

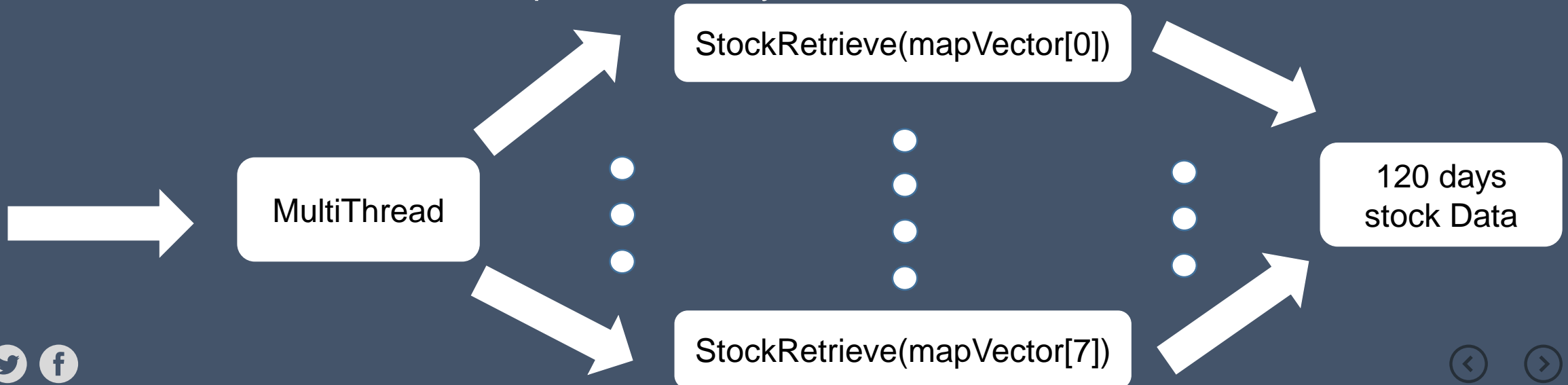
Calculation

- Grouping by ranking StockData.sur
- Calculate data member: sReturn, sAR, group, group_, reportIndex

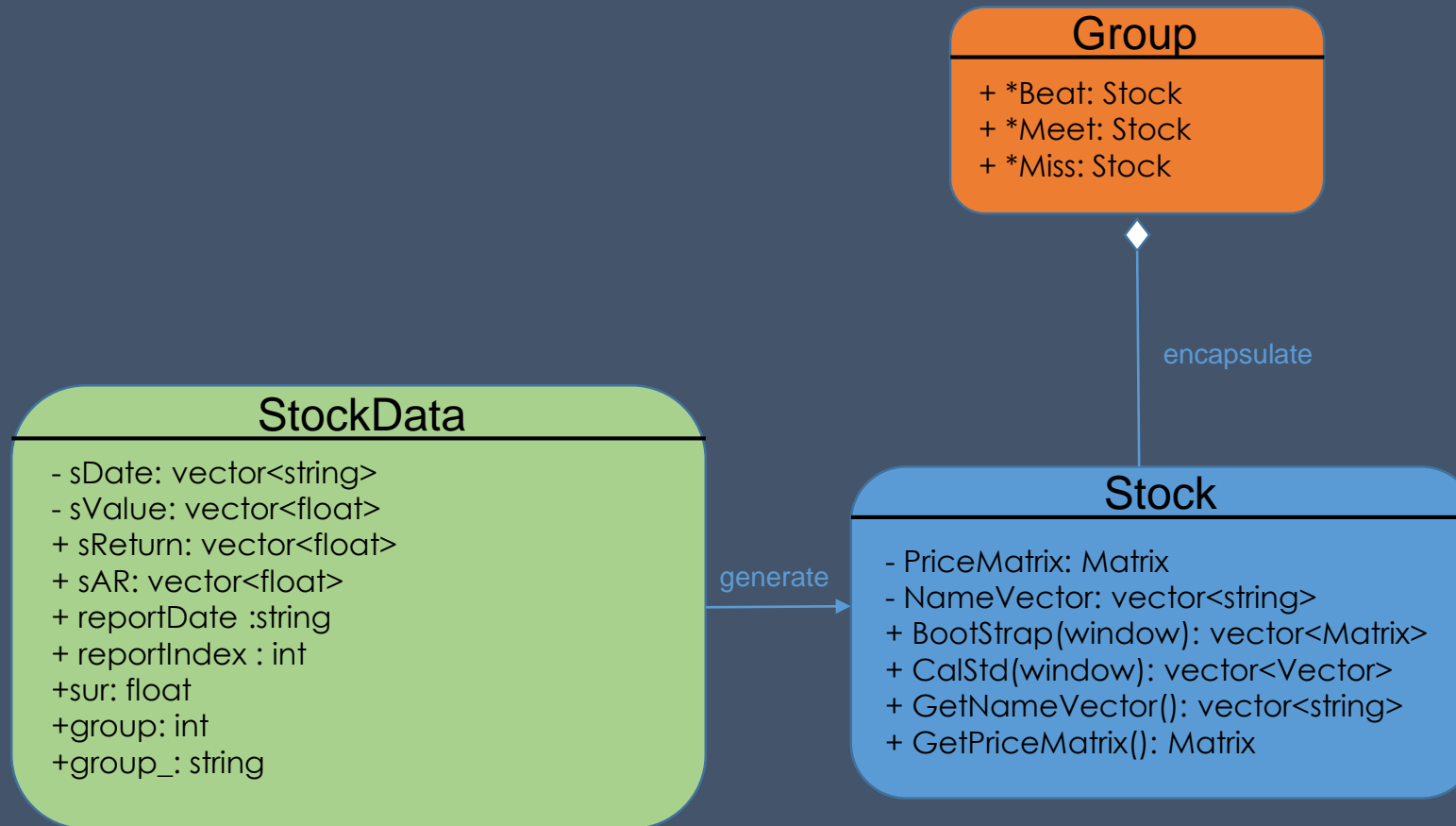


Multi Thread

- Multi threads are used to speed up data crawling from Yahoo Finance
- Overheads:
 - To avoid multi threads assessing the map at the same time
 - Copy the map and split it into 8 smaller maps
- Data crawling is expected to finished less than 20 seconds
 - 8 threads work on 8 smaller maps concurrently



UML



Class Structure

Stock.h

DATA TYPE	DATA MEMBER
MATRIX	PriceMatrix
VECTOR<STRING>	NameVector

MEMBER FUCNTION	RETURN TYPE	PARAMETER
BootStrap	VECTOR<MATRIX>	WINDOW
CalStd	VECTOR<VETCOR<FLOAT>>	WINDOW
GetNameVector	VECTOR<STRING>	VOID
GetPriceMatrix	MATRIX	VOID

Class Structure

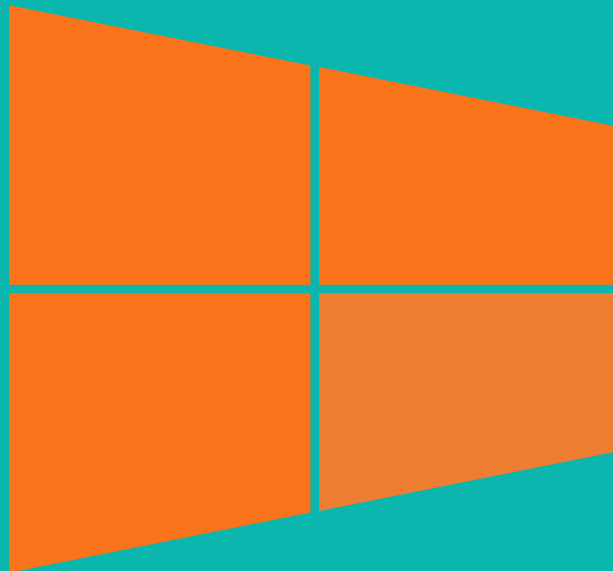
Group.h

DATA TYPE	DATA MEMBER
STOCK	* Beat
	* Meet
	* Miss

MEMBER FUCNTION	PARAMETER
Group	VECTOR<STOCKDATA>
~Group	VOID

Menu

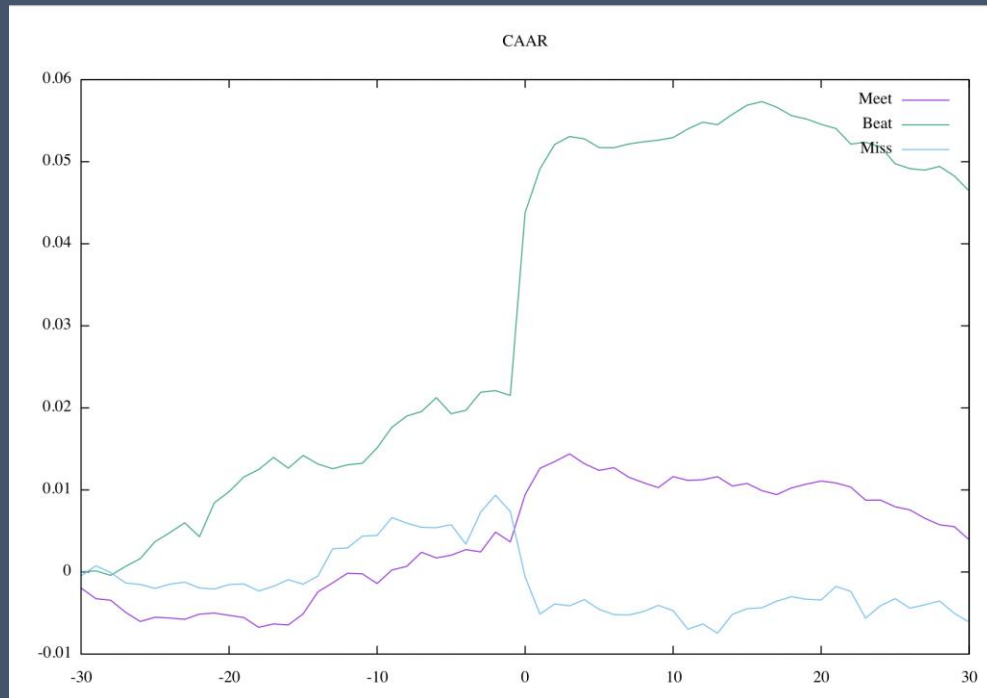
	DATA MEMBER	
Retrieve Stock Data	CalStd(int) -> ARR & STD	
	CalStd(int) -> CARR & STD	
	GetNameVector()	
Show Stock Historical Data	Show Beat Group	Stock Data
		Stock Name
	Show Meet Group	Stock Data
		Stock Name
	Show Miss Group	Stock Data
		Stock Name
Show AAR/CAAR of Each Group	Show AAR	
	Show CAAR	
Plot Graph of CAAR	plotresult(Matrix, int)	
Exit	0	



Result

Result

After the announcement, the difference between three groups increased a lot, which means the eps reporting will have a significant impact on stock price.



Beat

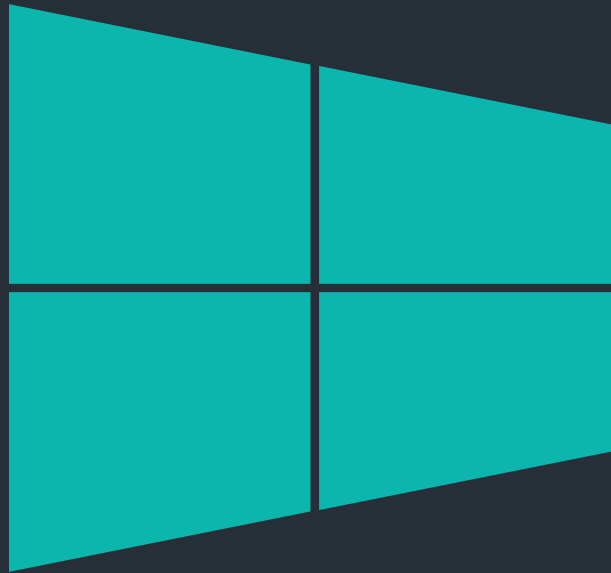
Increase

Meet

Modest Increase

Miss

Decrease



THANK YOU !

Keep It Simple & Awesome....