

## ***The Wild West of Programming***

*Josh Shvartsman  
Stern, Sophomore  
\_Member Presentation*

**NASDAQ**

1



## **Announcements**

- This Weekend:
  - Math Workshop → RSVP (Saturday)
- Intro to LaTeX thing next week
- Pay / Take Test / Swipe ID
  - 1<sup>st</sup> Semester = Trial Period
  - Either in or out at that point

**NASDAQ**

2



## Goals of QuantFS

- Professional Development
- Recruiting (Interviews)
- Education
  - Curriculum
  - Projects
- Legacy & Alumni



## Professional Development

- Next Semester
  - Mock Quant Interviews
  - General Tips
  - Interviewing Strategies
  - Job Postings (a lot of `em)



## Goals of QuantFS - Curriculum

- Finance
  - Options, Fixed Income, Credit, Equities
- Programming
  - Framework for Quant model, Develop efficient programming techniques, 2 Languages per year.
- Economics
  - Macro Indicators, Micro Effects, Develop Factor Idea for Yearly Model, Equities
- Mathematics
  - Linear Algebra, Statistics, Probability, Math of Finance, ...

**NASDAQ**

5



## Goals of QuantFS - Project

- Programming
  - Develop dynamic model for updating (bulk of the work)
- Finance + Economics
  - How to choose the factors, testing
- Mathematics
  - Portfolio optimization and math of the model-making

**NASDAQ**

6



## Legacy & Alumni

- We want to keep the relations
- Need a strong board
  - Need interested freshmen and sophomores to apply for mentees
  - Need juniors to lead

\*\*\*MUST SHOW INTEREST SOON\*\*\*

**NASDAQ**

7



```
void qfs_103007 {  
Why Programming?
```

- Programming doesn't sound like linear algebra or matrix math or integrals...so what the heck does it have to do with quant?
- Ever thought about those program trading models?
- Or those algorithms?
- How are they *implemented*?

**NASDAQ**

8



## What Programming?

- It depends.
- Applications
  - Algorithmic/Program Trading?
  - Excel VBA?
  - Repetitive Tasks?
  - Mother of All Quant Analysis?!?



## Language “genres”

- Assembly
- Scripting
- Compiled Languages
- Interpreted Languages
- Web Languages



## Assembly

- Lowest-level language available
- Complete pain in the arse
- Then why use it?
  - Speed → Less “compiler risk”
  - Control structures → Better control than simple if/then/else, for/next, etc.

```
Mov eax, 5
```

```
Mov ecx, "Hello"
```

```
Mov ebx, 1
```

```
Mov eax, 4
```

```
Int 0x80
```

Print “Hello”

**NASDAQ**

11



## Scripting

- Least powerful high-level languages
- Examples:
  - JScript
  - VBScript / VBA
  - Perl
- Use for simple, repetitive tasks
  - Stock quote retrieval
  - File manipulation
  - Easy math...

**NASDAQ**

12



## VBScript Example

```
dim x, y, z
x = cdbl(inputbox("Return #1 (%)" ))
y = cdbl(inputbox("Return #2 (%)" ))
z = cdbl(inputbox("Return #3 (%)" ))
dim average_vars
average_vars = (cdbl(x) + cdbl(y) + cdbl(z))/3
msgbox("The arithmetic average is: " & vbnewline &
    average_vars)
dim geom_avg
geom_avg = (((x+1) * (y+1) * (z+1)) ^ (1/3))-1
msgbox("The geometric average is: " & vbnewline &
    geom_avg)
```



## Compiled Languages

- The bread and butter, most common
- Examples
  - C++
  - VB6 (but not .NET)
- Use for most complex tasks
  - Platform-dependent (versus interpreted)
  - Trend analysis
  - Statistical analysis



## VB6 Example

```
Sub Main()  
Open "C:\values-x.txt" for input as #1  
Dim values-x() as double  
Redim values-x(0)  
Do until eof(#1)  
Line Input #1, values-x(ubound(values-x))  
Redim preserve values-x(ubound(values-x) + 1)  
Loop  
Close #1  
Open "C:\values-y.txt" for input as #1  
Dim values-y() as double  
Redim values-y(0)  
Do until eof(#1)  
Line Input #1, values-y(ubound(values-y))  
Redim preserve values-y(ubound(values-y) + 1)  
Loop  
Continued...
```

**NASDAQ**

15



## VB6 Example (cont.)

```
Dim sum-xy, sum-x, sum-y, sum-x2 as double  
For myforloop = 0 to ubound(values-y)  
Sum-xy = sum-xy + values-x(myforloop) * values-  
y(myforloop)  
Sum-x = sum-x + values-x(myforloop)  
Sum-y = sum-y + values-y(myforloop)  
Sum-x2 = sum-x2 + (values-x(myforloop)^2)  
Next  
Dim powerbar-slope, musclemilk-intercept as double  
Powerbar-slope = (ubound(values-y) * sum-xy - sum-x * sum-  
y)/(ubound(values-y)*sum-x2 - (sum-x^2))  
Musclemilk-intercept = sum-y / ubound(values-y) - powerbar-  
slope * (sum-x / ubound(values-y))  
MsgBox("y = " & powerbar-slope & "x + " & musclemilk-  
intercept)
```

**NASDAQ**

16



## Interpreted Languages

- The next generation → Cross-platform
- Examples
  - .NET (VB.NET, C#)
  - Java
- Applications
  - Rich user interfaces to complex back-ends
  - End-user apps for multiple platforms
  - Web applications

**NASDAQ**

17



## VB.NET Example

```
Dim rList() As MyApp = New MyApp(0) {}  
rList(0).appid = 0  
rList(0).AppName = "PlanAhead"  
Dim xInxstr As String, l2 As Short  
xInxstr =  
mwc.DownloadString("http://www.xilinx.com/ise/planahead/pa_download.htm")  
linkstart = InStr(xInxstr, " Install Kits")  
Dim xInxvdt As DateTime  
xInxvdt = DateTime.Parse(xInxstr.Substring(linkstart + 14, 8))  
rList(0).verDate = xInxvdt  
l2 = InStr(linkstart - 20, xInxstr, ">")  
planver = xInxstr.Substring(l2, (linkstart - l2 - 1)).Replace("PlanAhead ", "")  
rList(0).Version = planver  
Dim linkb As String =  
"ftp://ftp.xilinx.com/pub/applications/misc/planahead-"  
rList(0).Link = linkb + planver + "-win32.exe"  
Return rList
```

**NASDAQ**

18



## VB.NET Example (Output)

- Output from the program that uses this component:

```
Name: PlanAhead  
Version: 9.2.3  
Link: ftp://ftp.xilinx.com/pub/applications/misc/planahead-9.2.3-  
win32.exe  
Version Date: 9/27/2007 12:00:00 AM
```



## Web Languages

- I wonder what these are for...
- Examples
  - (X)HTML
  - CSS
  - PHP
  - ASP / ASP.NET
- Applications
  - A pretty web site explaining why your CDOs are suddenly illiquid and your return last month was -95%
  - Attracting more money into your multi-strategy fund based on movements of Pluto



## PHP Example

- Brought to you by that other investment club in Stern and QuantFS Grand Poobah Varun

```
$varunday = date("d");
$varunmonth = date("m");
$varunsecond = date("s");
$varunminute = date("i");
$varunhour = date("H")+5;
//I (varun) added 5 to our hours because I needed to account for EST -
500 (later on I have a form to test...
$varunyear = date("Y");
$FGHIJK =
    gmmktime($varunhour,$varunminute,$varunsecond,$varunmonth,
    $varunday,$varunyear);

$month = Date_Calc::dateFormat($day,$month,$year,"%B%n");
```

**NASDAQ**

21



## Conclusion

- Quants should know programming as well as math
- Depending where you find yourself, dictates what you should learn
  - Algorithms: Assembly and C++
  - Data mining: VBA and Perl
  - Web development: PHP, ASP, CSS, XHTML
  - General: Scripting (makes your life easier)

**NASDAQ**

22



```
Return true; }  
(The end)
```

**NASDAQ**

23

