Manager  Lauren O’Connor, leo2118
System Architect  Paddy Quinn, pmq2101
Language Guru  Josh Fram, jpf2141
Tester  Rob Cornacchia, rlc2160
Background & Goals
Development
Roles and Responsibilities

Manager
Lauren O’Connor, leo2118

System Architect
Paddy Quinn, pmq2101

Language Guru
Josh Fram, jpf2141

Tester
Rob Cornacchia, rlc2160

Paddy
compiler development (semantics)

Lauren
compiler development (syntax)

Josh
java libraries
YQL integration

Rob
test suite

not so strong
financial knowledge
strong

Lauren
strong

Paddy
not so strong
Testing

**Unit Testing**

Lexical and Syntactic Test

```plaintext
int x_1;
string w__;
print "success";
print "\n";
```

Semantic Test

```plaintext
function int main() {
    int x;
    x << 1 + "string";
};
```

**Regression Testing**

```plaintext
Compiling define_stock_test.finl ...
Running define_stock_test ...
define_stock_test passed.
Compiling div_assign_test.finl ...
Running div_assign_test ...
div_assign_test passed.
Compiling float_test.finl ...
Running float_test ...
float_test passed.
Compiling hello_world.finl ...
Running hello_world ...
hello_world passed.
Compiling if_decl_test.finl ...
Fatal error: exception Semantics.Except("Variable 'x' is already defined!")
Running if_decl_test ...
Error: Could not find or load main class if_decl_test
if_decl_test passed.
Compiling if_test.finl ...
Running if_test ...
if_test passed.
Compiling int_float_op_test.finl ...
Running int_float_op_test ...
int_float_op_test passed.
Compiling main_test.finl ...
Fatal error: exception Semantics.Except("Function 'main' is already defined!")
Running main_test ...
```

**Other Platforms Used**

```plaintext
$ menhir --interpret --interpret-show-cst parser.mly
$ ./finlc -a input.finl #prints ast
$ ./finlc -s input.finl #prints sast
```
Brief Tutorial
Example

```plaintext
portfolio "myPortfolio";

function int add(int x, int y) {
    return x + y;
};

function int subtract(int x, int y) {
    return x - y;
};

int two;
int one;
two << 2;
one << 1;
temp int result;
result = subtract(two, one);
print result; # prints 1
```
# Example declarations and assignments

```python
int x;
x <<= 12;

stock i_love_apple;
i_love_apple <<= @AAPL;

print @TSLA[epsEstimateCurrentYear];

order apple_order;
apple_order <<= 100 of apple;
```

```
print 3**2;

sell 50 of @AMZN;
buy my_order;

not(presentation=good)? {
    make_joke();
};

portfolio "my_portfolio_name"; print ;
```
Control Flow

**Conditionals**

Conditionals read as true/false questions
- ? is equivalent to if
- and ! equivalent to else

```
int number;
number <<= 0;

number > 0 ?
  {print 1;}
! {print 0;};
# this will print 0
```

**Loops**

- While loops can be used both as a conditional (while) and iterative (for) loop
- Using a while loop to iterate a set number of times, as a for loop, is shown below

```
int i;
i <<= 0;
while i<5 {
  print i;
  i += <<1;
};
# executes 5 times
```

- When loops are a key feature in finl
- Starts a new thread to support being able to place multiple limit orders
- They check a conditional periodically, and when it is satisfied, the body is executed once

```
when apple[FiftydayMovingAverage] > apple[TwoHundreddayMovingAverage] {
  buy myOrder;
}
```

# This order will only execute if the fifty day moving average goes over the two hundred day moving average
Unique Features

Printing Complex Data Types
ability to print stocks, orders, and portfolios

YQL Database Calls
abstraction of Yahoo Finance database calls from the user

When Loops
ability to periodically check conditional and work with multiple threads using when loops

CSV Portfolios
ability to import and export CSV portfolios to work over multiple sessions
Implementation
Java Libraries

**Stock**
- quotes subclass
- fundamental subclass
- dividends subclass
  - statistics, variables, etc.

**Portfolio**
- holding subclass
- account value
- print
- CSV import and export

**Order**
- stock
- size
- share price
- execution date
- whether it’s been executed
- order type (buy/sell)

**Lib**
- boolean manipulation
- string comparison
Demo & Questions