LISEIDY BUENO (LB3347)
KENYA PLENTY (KGP2111)
PAZIT SCHRECKER (PRS2143)
LINDSEY WALES (LBW2149)
KATRINA ZHAO (KZ2335)

FUNC-Y JAVA
<table>
<thead>
<tr>
<th>OVERVIEW</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Team</td>
</tr>
<tr>
<td>Motivation</td>
</tr>
<tr>
<td>Syntax</td>
</tr>
<tr>
<td>Program Structure</td>
</tr>
<tr>
<td>Architecture</td>
</tr>
<tr>
<td>Test Suite</td>
</tr>
<tr>
<td>Demos</td>
</tr>
<tr>
<td>Reflections and Advice</td>
</tr>
</tbody>
</table>
The Team

Katrina: Systems Architect
Thinks Java is too funky

Liseidy: Systems Architect
Thinks Java is too clunky

Pazit: Tester
Likes Java

Kenya: Manager
What's Java?

Lindsey + Chewy: Language Gurus
Think Java isn’t funky enough
MOTIVATION: JAVA IS A DIFFICULT FIRST LANGUAGE

• Java and Python are huge languages to begin with, especially when students are just learning about algorithms
• FUNC-y Java allows programmer to warm up to Java-like syntax and the concept of algorithms before diving into dealing with objects
### SYNTAX: PRIMITIVE TYPES

In FUNC-y Java, all first class objects – types that may be passed to and returned by functions – are considered primitive types*

<table>
<thead>
<tr>
<th>Type</th>
<th>Associated keyword</th>
<th>Definition</th>
<th>Examples (literals)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Integer</td>
<td>int</td>
<td>An integer constant is a sequence of digits.</td>
<td>57692 0 -928</td>
</tr>
<tr>
<td>Boolean</td>
<td>bool</td>
<td>Boolean values are stored as single-bit integer value, 0 or 1.</td>
<td>true false</td>
</tr>
<tr>
<td></td>
<td></td>
<td>represented in FUNC-y Java as the keywords ‘true’ and ‘false’</td>
<td></td>
</tr>
<tr>
<td>Float</td>
<td>float</td>
<td>A floating constant consists of a decimal part and a fractional part,</td>
<td>5.6 0.0000 3.14159</td>
</tr>
<tr>
<td></td>
<td></td>
<td>which are both mandatory. Every floating point constant is taken to</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>be double precision.</td>
<td></td>
</tr>
<tr>
<td>Character</td>
<td>char</td>
<td>A character constant is one ASCII character enclosed in single</td>
<td>‘c’ ‘%’ ‘9’</td>
</tr>
<tr>
<td></td>
<td></td>
<td>quotation marks</td>
<td></td>
</tr>
<tr>
<td>String</td>
<td>str</td>
<td>A string constant is a sequence of ASCII characters enclosed within</td>
<td>“funky” “” “ life on mars ”</td>
</tr>
<tr>
<td></td>
<td></td>
<td>double quotation marks</td>
<td></td>
</tr>
<tr>
<td>Cry</td>
<td>cry</td>
<td>Cry is FUNC-y Java’s void type</td>
<td>N/A: no variable may be defined of type cry</td>
</tr>
</tbody>
</table>
SYNTAX: OPERATORS

LEFT TO RIGHT ASSOCIATIVITY

Binary
- +, -, *, /: addition, subtraction, multiplication, and division for two ints or floats
- %: modulus operator for two ints
- and, or: logical and, logical or for two bools

Relational
- <, >, <=, =>: LT, GT, LEQ, GEQ on int and float

Equality
- !=, ==: not equal to, equal to for primitives

RIGHT TO LEFT ASSOCIATIVITY

Assignment
- =: assignment of a variable to any type
- +=, -=, *=, /=: mathematical binary operation and then reassignment to same variable on ints and floats
- %=: mod operation and then reassignment to same variable on ints

Unary
- -: makes a float or int the opposite sign
- ++, --: increments or decrements an int
- not: negates a bool
DATA STRUCTURES: ARRAYS

Declaration format
array<type, size> arrayName = [ ... ];

Elements may be accessed using an integer literal or integer type variable, allowing them to be iterated through using loops.

Arrays may contain types: int, str, float
Fixed size and of uniform type
Instantiated at time of declaration.

array<float, 2> fL = [3.4, 4.4];
float x = fL[0];
print(x);
• Declared with keyword “func” followed by primitive return type and unique function name.
• Functions are invoked with their name and arguments (if any).
• Each program requires a main() function that returns type int
• Functions may be declared in any order throughout the program, but each program only has access to the functions declared in the given file
• FUNC-y Java supports recursion through recursive functions

```java
func int fact(int n) {
    if (n <= 1) {
        return 1;
    } else {
        return n * fact(n-1);
    }
}
func int main() {
    print(fact(8));
    print(fact(0));
}
```
Primitive type variables may be declared an initialized in 1 of 2 ways:

1. Variables of primitive types may be declared in one statement and initialized with a value in a separate expression.

2. Variables may be declared and initialized in the same statement.

FUNC-y java supports local variable declaration and assignment throughout a function.

Primitive variable types may have values reassigned at any point in the program.
ARCHITECTURE

FUNC-y JAVA File → Scanner → Tokens → Parser → AST → Semant

Semantically Checked AST → LLVM IR → CodeGen

C Compiler → LLVM IR

FUNC-y JAVA Executable
TEST SUITE

• The automated testing suite for FUNC-y Java compares output of sample programs to expected output in .out and .err files

• Two types of tests:
  I. test-#: Passing tests print output to .out files and compare this to expected output
  II. fail-#: Failing tests print error messages to .err files and this is compared to expected error messages

• Unit tests were added by members when implementing a new feature. More unit tests and integration tests were written by the tester once changes were pushed to the main branch.
DEMOS

DEMO 1: STRINGS AND FUNCTIONS

“HI!”

DEMO 2: CHARS

‘L’

DEMO 3: ARRAYS

[2,4,6,8]
• It's easier to work by feature rather than by file
• Learn to love LLVM Moe
• Dream big and then be ready to be flexible on features
• Constant and clear communication is important
• Take advantage of TA and the Professor’s Office Hours!
• Plan ahead: the order you try to implement features really matters
• Pair programming is your friend, 5-person zoom calls with one screen are not