LISEIDY BUENO
(LB3347)

KENYA PLENTY
(KGP2111)

PAZIT SCHRECKER
(PRS2143)

LINDSEY WALES
(LBW2149)

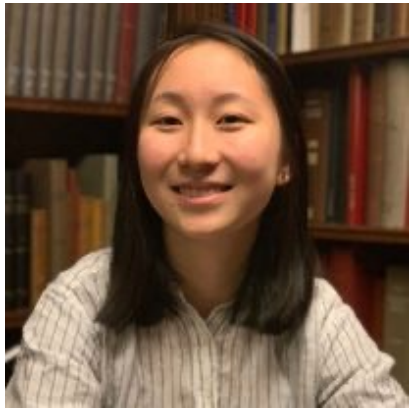
KATRINA ZHAO
(KZ2335)

FUNC-Y JAVA

OVERVIEW

- The Team
- Motivation
- Syntax
- Program Structure
- Architecture
- Test Suite
- Demos
- Reflections and Advice

THE TEAM



Katrina: Systems Architect
Thinks Java is too funky

Liseidy: Systems Architect
Thinks Java is too clunky

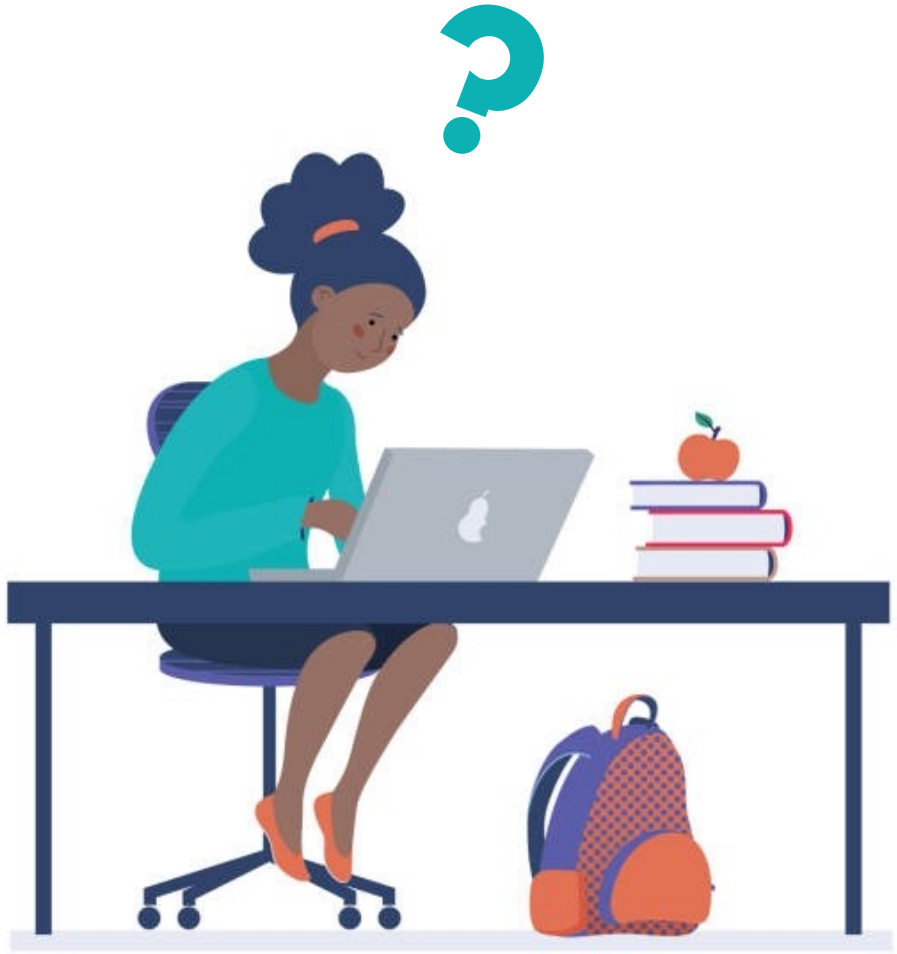
Lindsey + Chewy: Language Gurus
Think Java isn't funky enough

Pazit: Tester
Likes Java

Kenya: Manager
What's Java?

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

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

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

* Cry types are not technically primitive because they may not be passed but they may be returned

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);
```

PROGRAM STRUCTURE: FUNCTION DECLARATION AND INVOCATION

- 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

```
func int fact(int n) {
    if (n <= 1) {
        return 1;
    }
    return n * fact(n-1);
}

func int main() {
    print(fact(8));
    print(fact(0));
}
```

PROGRAM STRUCTURE: VARIABLE DECLARATION AND ASSIGNMENT

Primitive type variables may be declared an initialized in 1 of 2 ways:

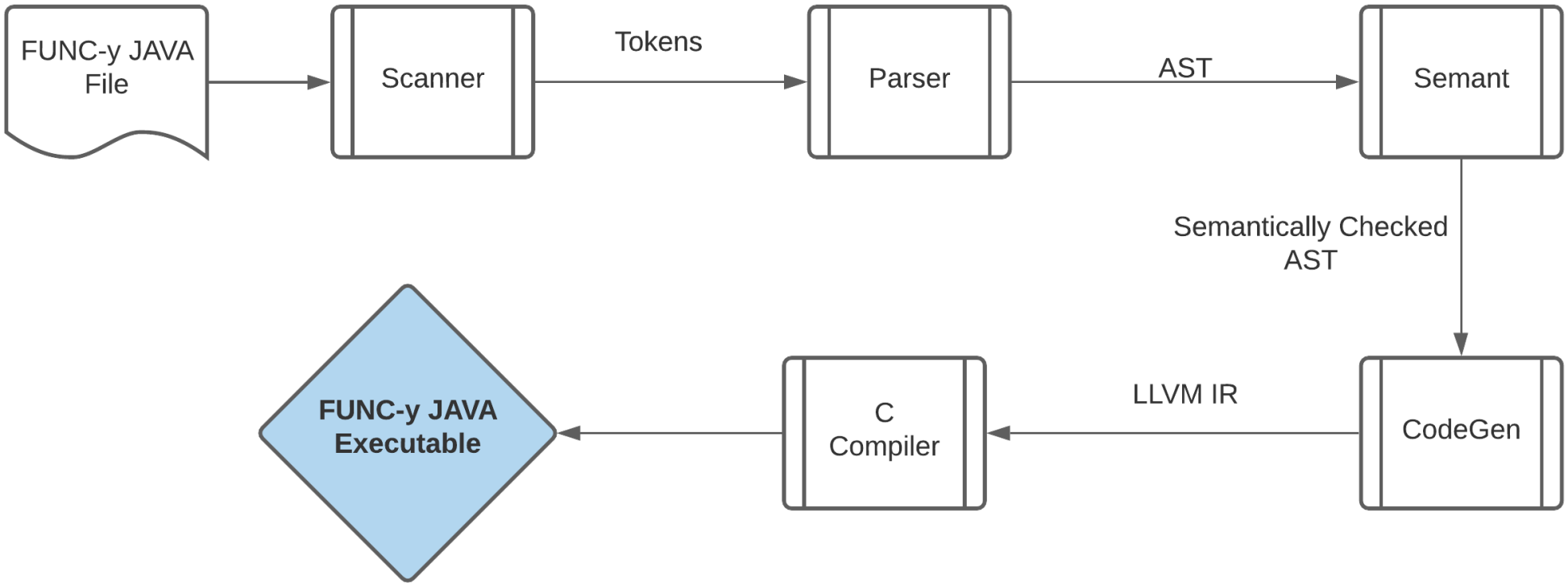
1. Variables of primitive types may be declared in one statement and initialized with a value in a separate expression.
2. Variables may be declared and initialized in the same statement.

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

```
func int main() {
    int a = 9;
    print(a);
    int b;
    int c = 10;
    b = a + c;
    print(b);
    a = 4;
    print(a);
    print(b);
}
```

ARCHITECTURE



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

DEMO 2: CHARS

DEMO 3: ARRAYS

REFLECTIONS AND ADVICE FOR FUTURE GROUPS

- 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