Option B

Programming Languages Final Project

...

Catherine Tu
Rolando Melendez

Swift Subset...

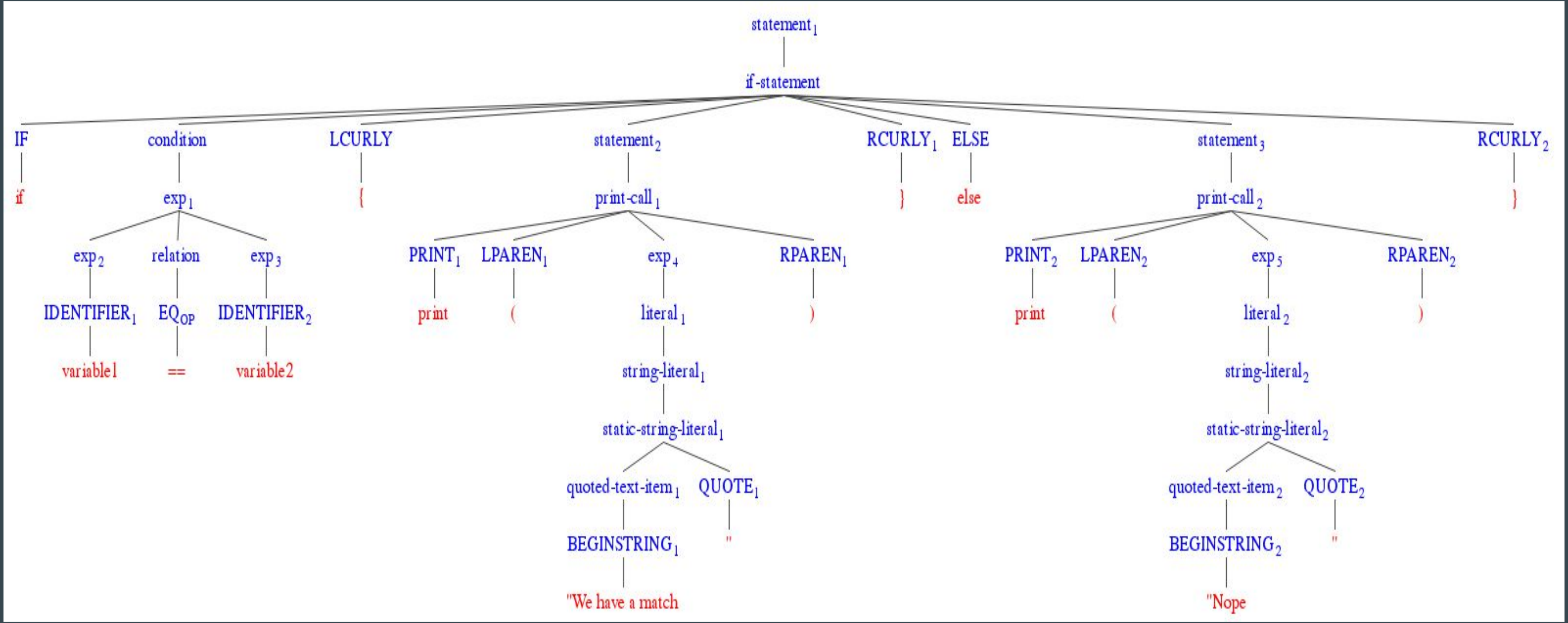
- Variable assignment
  - let x = 34
  - var y = 33+1
- Logical Statements
  - if x == y { print (“We have a match”) }
  - else { print (“Nope”) }
- Print Statements
  - print (“Anything”)
- Java Streams to Swift
  - Employee list data
  - Uses ‘exec’ function
- Python List Comprehension to Swift
  - Match the Java Stream Output
- Python Closure Implementation
  - Reads in CSV file of student data
  - Uses student class and major subclasses to create list of student objects
  - Filters for business students

Test Code...

```
let variable1 = (exec 'from java.lang import Math; toReturn = Math.max(23, 34)') + 1
let variable2 = 35

var variable3 = (exec 'from java.lang import Math; toReturn = Math.max(23, 34)') + 1
var variable4 = 99

if variable1 == variable2 { print("We have a match") } else { print("Nope") }
if variable4 < 40 { print("\(variable4) is less than 40" )} else { print("\(variable4) is not less than 40")}


//Demonstrate with java file stream 1
var list1 = (exec 'import Stream; toReturn = Stream.stream1(Stream.createEmpList())')

//Demonstrate with java file stream 2
var list2 = (exec 'import Stream; toReturn = Stream.stream2(Stream.createEmpList())')

var list3 = listcomp(list1)

print (list3)

if list2 == list3 { print("The output of the java stream and the output of the python list comprehension are the same.") } else { print("The outputs are
different.")}
print ("The output from the java stream: \(list2)")
print ("The output form the python list comprehension: \(list3)")

var list4 = (exec 'import Closure; Closure.createStudentsList(); toReturn = Closure.returnAllStudents()')
print(list4)
list4 = (exec 'import Closure; Closure.createStudentsList(); toReturn = Closure.returnAllBusinessStudents()')
print(list4)
```

Parse Tree...

Example: if variable1 == variable2 { print("We have a match") } else { print("Nope") }



AST Structure...

- Print statements:
  - print (“Hello World”)
  - AST is: ['print', 'Hello World']
- Declaration/Assignment:
  - let testOne = “Hello World”
  - AST is: ['let', 'testOne', 'Hello World']
  - Var testTwo = “Hello World”
  - AST is: ['var', 'testTwo', 'Hello World']
- If/Else:
  - if 1 == 2 { print ("This is a match") } else { print ("This is NOT a match") }
  - AST is: ['if', ['==', 1, 2], ['print', 'This is a match'], ['print', 'This is NOT a match']]
- Output from above:
  - This is NOT a match

Demo…

Our Bugs... 🐞🐛🐜🐝

1. Token for string literal difficult to differentiate from identifier token.
   a. Used alternative “hack” to print combinations of variable and string literals
   b. Note: Can only use one “\(expression)”
2. List Comprehension output includes unicode ‘u character that should not be present.

... Probably many more

Questions...?

Thank You!