JavaLite

Frances Cao (fc2679)
Mateo Maturana (mm5589)
Hongfei Chen (hc3222)
Ian Chen (yc3936)

April 26, 2021

Contents

1 Introduction **3**
1.1 Goals . . . 3
1.1.1 Syntax Simplification . . . 3
1.1.2 String and Array Improvements . . . 4

2 Language Tutorial **4**
2.1 Docker Configuration . . . 4
2.2 Compilation and Execution . . . 5

3 Language Reference Manual **5**
3.1 Lexical Structure . . . 5
3.1.1 Comments . . . 5
3.1.2 Identifiers . . . 5
3.1.3 Keywords . . . 6
3.1.4 Separators . . . 6
3.1.5 Operators . . . 6
3.1.6 Literals . . . 6
3.2 Types and Variables . . . 7
3.2.1 Primitive Types . . . 8
3.2.2 Objects . . . 9
3.2.3 Variables . . . 9
3.2.4 Strings . . . 10
3.2.5 Arrays . . . 10
3.3 Statements and Expressions . . . 11
3.3.1 Statements . . . 11
3.3.2 Expressions . . . 12
3.3.3 Operator Precedence . . . 13
3.3.4 Functions . . . 13
3.4 Classes . . . 14
3.4.1 Dot Operator . . . 14
3.5 Built-In Functions . . . 14
3.5.1 Arrays . . . 15
3.5.2 Strings . . . 15
3.5.3 Print . . . 16
3.6 Syntax/Style Guide . . . 16

4 Project Plan **17**
4.1 Planning Process . . . 17
4.2 Specification Process . . . 17
4.3 Development Process . . . 17
4.4 Testing Process . . . 18
4.5 Roles and Responsibilities . . . 18
4.6 Project Timeline . . . 19
4.7 Project Log . . . 19
4.8 Software Development Environment . . . 19
4.9 Programming Style Guide . . . 19

5 Architectural Design **20**
5.1 Block Diagram . . . 20
5.2 Scanner . . . 20
5.3 Parser and AST . . . 20
5.4 Semantic Checking . . . 21
5.5 Code Generation . . . 21
5.6 C Libraries . . . 21

6 Test Plan **22**
6.1 Representative Programs . . . 22
6.2 Testing Tools . . . 23
6.3 Automated CD/CI . . . 24
6.4 Performance Benchmarking . . . 24
6.5 Target Language Program (LLVM) . . . 25
6.5.1 GCD . . . 25
6.5.2 Bubble Sort . . . 26
6.5.3 Class . . . 30

7 Lessons Learned **33**
7.1 Frances Cao . . . 33
7.2 Mateo Maturana . . . 33
7.3 Hongfei Chen . . . 33
7.4 Ian Chen . . . 34

8 Appendix **34**
8.1 scanner.mll . . . 34
8.2 parser.mly . . . 36
8.3 ast.ml . . . 38
8.4 semant.ml . . . 41
8.5 sast.ml . . . 49
8.6 codegen.ml . . . 52
8.7 javalite.ml . . . 61
8.8 stringfuncs.c . . . 62
8.9 Makefile . . . 64
8.10 testall.sh . . . 65

8.11 Testing Files 69
8.11.1 Test Cases 69
8.11.2 Fail Cases 83
8.11.3 Performance Tests 89
8.11.4 Demo Files 91

9 Project Log **93**

1 Introduction

JavaLite is a multi-paradigm language that allows the user to freely use both functional and object-oriented paradigms. While Java is one of the most popular programming languages for beginners, it can be challenging for new programmers to get familiar with its excessive syntax and strict object-oriented programming rules. In contrast, JavaLite has simplified syntax compared to Java. Further, JavaLite supports more built-in functionality for non-primitive data types, including String and Array. These built-in methods are inspired by Python and the goal is to provide a more intuitive way of String/Array manipulation.

1.1 Goals

1.1.1 Syntax Simplification

In Java, each program has to have at least the same basic outline.

```
public class Program {
  public static void main(String[] args) {
    // main method begins here
  }
}
```

For new programmers, this is often confusing and can lead to many questions that are usually not answered until much later in an introductory class such as public vs private, static methods, method arguments, and arrays. This is mainly because Java is completely object-oriented, so every method must be part of a class, including the main method. JavaLite removes the need for this complicated syntax by allowing functional program. Therefore, there is no main method and no class is needed to execute any code, similar to JavaScript.

Additionally, consider the default print method in Java: *system.out.print()*. The only relevant part to a new programmer is print since they won't initially understand the purpose behind system.out. In JavaLite, printing is done with *print()*.

1.1.2 String and Array Improvements

In Java, Strings and Arrays present many unique issues that may be confusing to new programmers. With strings, we cannot directly compare strings with $<, >, <=$, or $>=$. Instead in Java we have to use *compareTo()*, however this may be confusing to new programmers so these binary operations are implemented in JavaLite. Also, in Java == is not accurate in comparing the values of two strings so *equals()* must be used instead. This problem is resolved in JavaLite so string comparisons remain consistent with primitive types such as int.

One simple change that we make is array literals. Although arrays are declared with *type []*, arrays literals are declared with a list of values enclosed by {}. This doesn't make a lot of sense, so instead JavaLite array literals are list of values enclosed by [].

2 Language Tutorial

Some familiarity with git and bash/shell is assumed. First the repository must be cloned into your local machine using *git clone*.

```
$ git clone git@github.com:FranCao/javalite.git
```

2.1 Docker Configuration

To run JavaLite, Docker must be installed on your local machine. To install Docker, refer to Docker's documentation.

1. Navigate to the root of the JavaLite repository. If you haven't changed directory since you cloned, you can run the following command.

   ```
   $ cd javalite
   ```

2. Make sure docker is running correctly. To test, you can run

   ```
   > docker run hello-world
   ```

   If this fails, you will need to correct your installation.

3. Next, invoke the docker image.

   ```
   > docker run --rm -it -v `pwd`:/home/javalite -w=/home/javalite
       ↪ columbiasedwards/plt
   ```

4. Finally, if you're not in the correct directory, run *cd javalite* to switch to the correct directory.

2.2 Compilation and Execution

To compile the JavaLite compiler and run JavaLite's test suite, you can use the following command.

```
make
```

Otherwise, you can also just write, compile, and execute a single program. First, the JavaLite compiler must be compiled. To do this, run the following commmand.

```
make all
```

The following sample code is a hello world program in JavaLite.

```
print("Hello, World!");
```

Save this code into a *helloworld.jl* file. To compile and run this code, run the following commands in your terminal.

```
$ ./BuildAndRun helloworld.jl
Hello, World!
```

3 Language Reference Manual

3.1 Lexical Structure

3.1.1 Comments

JavaLite supports both single line and multi-line comments.

```
// This is a single line comment
```

A single line comment starts with "//" and ends at the end of the line. Any ASCII characters before the end of the line are considered comments.

```
/* This is a
muli-line
comment */
```

A multi-line comment is wrapped by "/*" and "*/". Namely, any ASCII characters after "/*" are considered comments until there is a "*/".

JavaLite does not allow nested comments. The comments will be ignored by the compiler and will not be executed.

3.1.2 Identifiers

An identifier in JavaLite is an unlimited-length character sequence consisting of letters and digits "a-zA-Z1-9", except for those reserved for keywords (3.1.3). Note that identifiers must begin with letters.

```
// valid identifiers
javalite
name
person12

// invalid identifiers
14
1java
```

3.1.3 Keywords

There are a total number of 12 keywords reserved in JavaLite. Each keyword is a character sequence consisting of ASCII letters. The following are all the keywords.

bool?	else	int
string	for	void
class	if	while
double	return	null

3.1.4 Separators

There are 9 separators in JavaLite. They are tokens formed from ASCII characters. The following are all the separators.

(	]	;
)	{	,
[	}	.

3.1.5 Operators

There are 15 operators in JavaLite. They are tokens formed from ASCII characters. The following are all the operators.

=	-	/
==	<	&&
+	<=	||
>	$\star$	!=
>=	%	!

3.1.6 Literals

Literals in JavaLite represent any constant value of a primitive type (int, double, boolean) or the string type.

Int Literals
An integer literal is either the single ASCII digit 0, representing the integer zero, or any ASCII digit between 1 and 9, optionally followed by any sequence of ASCII digits between 0 and 9.

The int data type is a 32-bit signed primitive data type. The valid range of the integer literal is between $-2,147,483,648 (-2^{31})$ and $2,147,483,647 (2^{31}-1)$.

If the integer literal is larger than $2,147,483,647$ or less than the unary minus operator $2,147,483,648$, it will result in a compile-time error.

Examples of int literals:

```
0
-5640
42014
```

Double Literals
A floating point literal has a whole number, a decimal, followed by another whole number.

In decimal float literals, at least one digit, and either a decimal, an exponent, or a float type suffix is required..

Floating point literals are always of type double.

Examples of floating point literals of double type:

```
.23
3.145
0.0
-1.3
```

Boolean Literals
The Boolean type is represented by the literals true and false.

String Literals
A string literal is any sequence of zero or more characters enclosed in double quotes. Characters such as newlines may be represented by escape sequences.

Examples of string literals:

```
""   // the empty string
"\""
"Hello World!"
"This is a string literal\n that spans two lines"   // this forms a string with
    ↪ two lines of text
```

3.2 Types and Variables

JavaLite is a statically typed language. All variables and expressions will have a known type at compile time. JavaLite is also a strongly typed language, and limits the values that a variable can hold. Variables must be assigned to a value when declared.

3.2.1 Primitive Types

JavaLite support three primitive data types:

- int (number, 4 bytes)
- double (float number, 8 bytes)
- boolean (true or false, 1 byte)

Int Operations
JavaLite provides a number of operations for ints.

The first type of operators are the comparison operators, which result in a value of type int representing the boolean values. 0 indicates false while 1 indicates true. These operators are $<, <=, >, >=, ==, !=$.

```
2 < 4;    // 1
5 <= 4;   // o
0 > -10;  // 1
3 >= 3;   // 1
2 == 0;   // 0
2 != 0;   // 1
```

The second type of operators are the numerical operators, which result in a value of type int. These operators are $+, -, *, /, \%$. Note that / returns values of type int, not double, so the result will be the floor of the division.

```
10 + 2;   // 12
-8 - 10;  // -18
2 * 3;    // 6
4 / 5;    // 0
```

Double Operations
JavaLite provides a similar number of operations for doubles.

The first type of operators are the comparison operators, which result in a value of type int. These operators are $<, <=, >, >=, ==, !=$.

```
2.3 < 2.1;     // 0
5.0 <= 4.3;    // 0
0.0 > -.4;       // 1
3.2 >= 3.2;    // 1
0.1 == 0.2;    // 0
2.10 != 2.100; // 0
```

The second type of operators are the numerical operators, which result in a value of type double. These operators are $+, -, *, /$.

```
10.3 + 4.7; // 16.0
-8.0 - 10.0;    // -18.0
2.4 * 3. 0;    // 7.2
4.0 / 5.0;      // 0.8
```

Boolean Operations

JavaLite provides two relational operators for booleans, which result in a value of type int. These are != and ==.

```
false != false;  // 0
true != false;   // 1
false == false;  // 1
true == false;   // 0
```

JavaLite also provides three conditional operators for booleans, which result in a value of type int. These are !, ||, and &&.

```
!false;         // 1
!true;          // 0
true || true;   // 1
true || false;  // 1
false || false; // 0
true && true;   // 1
true && false;  // 0
false && false; // 0
```

3.2.2 Objects

Aside from primitive types, JavaLite contains three kinds of reference types (class types, type variables, and array types). Class types have a type name. If arguments appear in a class type, then it is a parameterized type. Objects are a class instance or an array. Reference values are pointers to these objects, along with a special null reference that refers to no object.

3.2.3 Variables

Variables in JavaLite provide a storage location that is named. A variable in JavaLite must have a specific type to determine the size and layout of its memory. Once a type is declared, the value held in the variable must be of that exact type. A variable is declared using the operator '=' and must be declared with a name of a valid string type.

```
// valid variable declarations
int x = 1000;
string s = "s";

// invalid variable declarations
int = 3;
4;
int x = 4.5;
int y;
char c = "string";
bool b = 2;
```

Each variable be initialized with a value.

3.2.4 Strings

JavaLite uses the String class to create and manipulate strings. A string object has a constant value and represents sequences of Unicode characters. String literals are references to instances of a string object. Strings in JavaLite are mutable. Strings can be declared like follows.

```
string str = "Hello, World!";
```

String Operations

JavaLite provides six comparison operators for strings, which result in a value of type int. These are !=, ==, <, >, <=, >=.

```
"str" != "stra";  // 1
"hi" == "Hi";     // 0
"a" < "c";        // 1
"z" > "y";        // 1
"ab" <= "aab"     // 0
"aaaab" >= "aaaab"// 1
```

JavaLite also provides the ability to use the '+' operator to concatenate two strings.

```
"str" + "ing";    // "string"
```

3.2.5 Arrays

An array in JavaLite is an object, and is assigned to variables of an object type. An array contains zero or more elements, in which the array is said to be empty in the case of zero elements. The elements in an array have no names and are referenced by array access expressions that use positive integer index values. In an array with n elements, with n being the length of the array, the elements of the array are referenced using integers from 0 to n-1.

An array type in JavaLite is written as the name of the elements type followed by one or more empty pairs of square brackets [], determining the depth of array nesting. All elements in an array are of the same type. If the components in an array are of type T, then the type of the array is written T[]. Elements in an array may be of any primitive or reference type. Arrays can be declared like follows.

```
string[] arr = ["hello", "world"];
```

Additionally, we can declare two-dimensional arrays as follows.

```
int[] arr1 = [1, 2, 3];
int[] arr2 = [4, 5, 6];
int[][] arr = [arr1, arr2];
```

Array Operations

JavaLite also provides the ability to use the "[]" operator to access elements of an array.

```
int[] arr1 = [3, 4, 5];
int[] arr2 = [10, 3, 2];
arr1[2];      // 4
arr2[0];      // 10
```

3.3 Statements and Expressions

3.3.1 Statements

Statements in JavaLite are of the following forms:

- If-Else Statements
- While Statements
- For Statements
- Expressions
- Return Statements

If-Else Statements
If statements in JavaLite allow for the conditional execution of a statement.

```
if (expression) {
  statement1;
}
else {
  statement2;
}
```

The expression must be of type boolean, or a compile-time error occurs.

In this case, the first statement will execute if expression evaluates to true and otherwise, the second statement will be executed. The conditional must be wrapped in a () and the statement to be executed must be wrapped in {}. However, a single line statement does not need to be wrapped in {}.

It is also possible to have standalone if statements or if-else if blocks without the trailing else.

```
if (expression) {
  statement;
}
```

Here, statement would be executed if expression evaluates to true. Nothing is executed if the expression evaluates to false.

While Statements
While statements in JavaLite execute an expression and a statement repeatedly until the value of the expression evaluates to false.

```
while (expression) {
  statement;
}
```

The expression must be of type boolean, or a compile-time error occurs. Similar to If-Else Statements, the expressions must be wrapped in () and the statements must be wrapped in {} (unless one line).

For Statements

For statements in JavaLite execute an initial statement and then executes an expression, statement, and update statement until the expression evaluate to false.

```
for (initial; expression; update) {
  statement;
}
```

The expression must be of type boolean, or a compile-time error occurs.

Return Statements

In JavaLite, return statements are used in functions (3.3.4) to return control (and data in some cases) to the invoker of a method.

```
T fun() {
  return expression;
}
```

Expression must be of type T, or a compile-time error occurs. In return statements, the expression is first evaluated then returned. For example,

```
int fun() {
  int x = 2;
  return x + 1;
}
```

In this case, $x + 1$ will be evaluated first, so the function will return 3.

3.3.2 Expressions

Expressions in JavaLite are a type of Statement that are of the following three syntactic forms:

- Primary Expressions
- Unary Operator Expressions
- Binary Operator Expressions

Primary Expressions

Primary Expressions in JavaLite are of the form of literals.

```
2;     // evaluates to 2
true;  // evaluates to true
```

Operation Expressions

Operator Expressions can be of unary or binary form. Operators specific to each of the types provided by JavaLite are discussed in 3.2.1, 3.2.1, 3.2.1, 3.2.4, and 3.2.5.

Assignment Operator

In JavaLite, the assignment operator '=' stores values into variables. This expression is evaluated right-to-left.

```
T var = expression;
```

The expression is first evaluated then stored into the variable of type T with name var. If the variable and evaluated value of the expression are not of the same type, a compile-time error occurs.

3.3.3 Operator Precedence

JavaLite executes operators with different precedence. The following table details the different levels of precedence with the highest levels of precedence at the top.

Operator	Description	Associativity
[] . ()	Array Access Object Member Access Parentheses	Left
!	Not	Right
* /	Multiplication Division	Left
+ − +	Addition Subtraction Array/String Concatenation	Left
<, <=, >, >=	Comparison	Left
==, !=	Equality	Left
&&	Logical AND	Left
||	Logical OR	Left
=	Assignment	Right

3.3.4 Functions

Functions in JavaLite are standalone. Functions take in a list of arguments and return one value.

```
T fun(R arg1, V arg2, ...) {
  /* do something */
  return expression;
}
```

Expression must be of type T. If not, a run-time error will occur. For example, we can write a function that increments values by 5 based on a boolean condition.

```
int fun(int x, bool b) {
  if (b) {
    x =  x + 5;
  }
  return x;
}
```

Functions are called by calling its name and providing the required number of parameters. A mismatch in the evaluated type of parameters or number of parameters will result in a compile-time error.

```
int fun(int x) {
  return x + 5;
}

fun();      // compile-time error
fun("hi");  // compile-time error
fun(2.3);   // compile-time error
fun(5);     // evaluates to 10
```

3.4 Classes

JavaLite supports classes with variables. Classes can be declared using the following syntax.

```
class Test {
  int x;
}
```

The constructor is called when a object is initialized. For example,

```
x = Test(3);
```

An object of type Test and name x would be initialized by passing one argument of value 3 into the constructor function. The constructor is assumed to be of return type void.

3.4.1 Dot Operator

In JavaLite, the dot operator, '.', is used to access values of an object (an instance of a class). These values are in the form of variables The dot operator is used in the following format.

```
x.var;
```

3.5 Built-In Functions

JavaLite supports built-in array structure with dynamic length similar to Python.

3.5.1 Arrays

arr[n]
arr[n] returns the nth element of the array.

```
int[] arr = [1, 2, 3];
arr[0];  // 1
arr[1];  // 2
```

int length(T[] arr)
length(T[] arr) returns the number of elements in arr.

```
int[] arr = [1, 2, 3];
length(arr);   // 3
```

3.5.2 Strings

string reverse(string str)
reverse(str) returns a string in the reverse order of str.

```
string str = "hello";
str = reverse(str);
// str = "olleh"
```

string upper(string str)
upper(str) converts all characters of str to upper-case.

```
string str = "hello";
str = upper(str);
// str = "HELLO"
```

string lower(string str)
lower(str) converts all characters of str to lower-case.

```
string str = "HeLLO";
str = lower(str);
// str = "hello"
```

string substring(string str, int a, int b)
substring(str, a, b) returns the substring of str between index a (included) and b (excluded).

```
string str = "hello";
string str_sub = substring(str, 0, 2);
// str_sub = "he"
```

- If a < 0, a compile-time error will occur
- If b $<$ a, a compile-time error will occur
- If b $>$ len(str), a run-time error will occur

int indexOf(string str, string c)

indexOf(str, c) returns the index of the first occurrence of c in str. If there is no occurrence of c, -1 is returned.

```
string str = "hello";
int a = str.indexOf(str, "h");
int b = str.indexOf(str, "l");
int c = str.indexOf(str, "k");
// a = 0, b = 2, c = -1
```

int len(string str)

len(str) returns the number of characters in str.

```
string str = "hello";
int x = length(str);
// x = 5
```

string concat(string str1, string str2)

concat(str1, str2) returns the concatenated string of str1 to str2.

```
string str1 = "hello";
string str2 = " world";
string str = concat(str1, str2)
// str = "hello world"
```

3.5.3 Print

JavaLite uses print to print strings to the terminal/console. Print can accept literals and variables of any type.

```
print("hi");  // "hi"
print(10);    // "10"
int x = 2;
string s = "hello";
print(x);     // 2
print(s);     // "hello"
```

- If print is called with more than one argument, a compile-time error will occur
- If print is called with any type other than int, double, boolean, or string, a compile-time error will occur

3.6 Syntax/Style Guide

- Any line that is not the beginning or end of a bracket block must end with a semi-colon. If not, a compile-time error will occur.
- All opening symbols must be matched with a closing symbol. More specifically, (), {} and [] must be matched. If not, a compile-time error will occur.
- Each line should be a maximum of 80 characters wide.

- Proper indentation of 4 spaces should be used, however indentation is ignored by the compiler.

4 Project Plan

4.1 Planning Process

The main goals and milestones are set during the first few project meetings, which were decided in consultation with Professor Edwards and T.A. Jianan Yao. Throughout the evolution of the project, we adjusted the timeline of individual tasks based on the progress. We have regular weekly group meetings to share progress and discuss next steps, as well as additional meetings within the group and with T.A. when encountering roadblocks during development.

4.2 Specification Process

The motivation of our language is to create a Java-like language but with simplified syntax and semantics, so that it is more user-friendly for beginner programmers. We want our language to incorporate the basic and key functionality of the Java language, as well as to accommodate the programming and learning needs of beginners. Therefore, all the features we designed for the language are based on such a purpose. We first determined the data types, such as integer, strings, and arrays, that are commonly seen in Java and other programming languages. Next, as a key part of any object-oriented language, we defined the syntax for class. To better emulate Java, a class defined in our language also has a constructor. Finally, in order for the language to be more beginner-friendly, we borrowed the idea from Python, so that the programmer does not need to define a main function to compile their program. We wrote the language reference manual based on these ideas and also added some helpful built-in functions for string and array operations. After turning in our LRM, a few adjustments were made to the specifications as the situation called for it, but no notable changes that diverged from our main purposes.

4.3 Development Process

We divided each individual task feature wise. For the development of each feature, we followed the stages of the compiler architecture. In other words, we defined the syntax part of the feature first, including scanner.mll, ast.ml and parser.mly, which is followed by the semantic checking part, semant.ml and sast.ml. We would use ocamlyacc and menhir commands to check for any parsing error. For the semantic part, we wrote test cases that covered both success and failure situations and used the pretty print functions to print out the typed version of the test programs. After thoroughly testing the parts stated above, we moved on to the code generation part of the feature (*codegen.ml*). Often, we would use the clang command to generate a LLVM file for a C program that

realizes the feature we were implementing and use that as a reference for writing the corresponding OCaml code.

4.4 Testing Process

The test cases are written throughout the project development process along the implementation of each individual feature. The tests for failure cases are mostly done while defining the syntax and semantics, which check whether the expected error message occur when using syntax that is not compatible with our language. The tests for successful cases were written after the code generation part was completed for each feature. Every aspect and possible usage of a feature are examined through these tests. More details about testing are covered in Section 6 Test Plan.

4.5 Roles and Responsibilities

Team Member	**Role**
Frances Cao	System Architect
Mateo Maturana	Language Guru
Hongfei Chen	Manager
Ian Chen	Tester

Below are the features and the group members who are responsible for each feature:

Arrays	Hongfei, Frances, Mateo
Built-In Functions	Frances
Classes	Hongfei
Pretty-printer	Hongfei, Frances
Main Dependency Removal	Ian, Mateo, Hongfei
Print (All types)	Frances, Hongfei
Variable Assignment	Mateo, Hongfei
String Binops	Hongfei, Frances

Testing and the final report were completed by every group member. Mateo was in charge of typesetting and documentation for the final report and LRM.

4.6 Project Timeline

Date	Milestone
1/23	Initial Meeting
1/24	JavaLite Proposal Idea
1/25	Proposal Finalized
2/13	Initial Scanner + Parser Work
2/19	LRM Draft
2/20	LRM Finalized + Scanner/Parser Work
2/23	TA Meeting
2/24	Finalized Parser + Scanner
3/15	TA Meeting
3/21	TA Meeting
4/18	Meeting to Establish Deliverables for Final Week
4/21	Final Week Check in
4/24	Meeting for Presentation Layout
4/26	Final Meeting; Presentation and Report

4.7 Project Log

Our project log is included in Section 9.

4.8 Software Development Environment

The main programming language that we used was OCaml with the LLVM library. OCamlyacc and OCamllex extensions were used for compiling the scanner and parser front end. We used C language to build some of the built-in functions and GCC was used to compile and link the C files. Each member used different development environments, but the most commonly used one was Visual Studio Code with its OCaml Platform extension.

4.9 Programming Style Guide

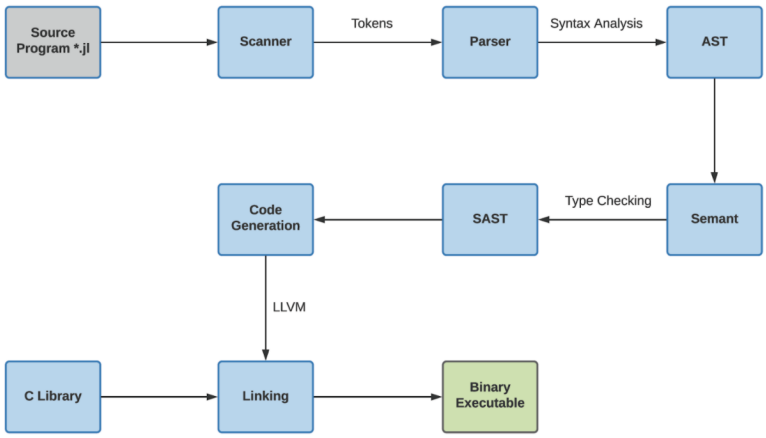
The general guidelines we followed while programming are as followings:

- Formatting: We followed the OCaml formatting and editing style. We made sure the code is indented properly so that the scopes would be understandable to the readers. Blocks of codes that are implemented for different features are separated by empty lines. The naming convention we used for variables is variable_name style (as opposed to camelCase).

- Commenting: Comments were included for each block of code explaining its general implementation idea and pointing out the corresponding feature for which it is used.

5 Architectural Design

5.1 Block Diagram



5.2 Scanner

File: *scanner.mll*

The scanner takes in a Javalite source program and translates it into a stream of tokens for identifiers, keywords, operators, and literals. Comments and whitespaces are ignored at this stage, and there is support for both single line and multiline comments. If any characters are found to be illegal, the scanner will throw an error. The tokens from the scanner are read in by the parser next to create the Abstract Syntax Tree.

Frances and Hongfei implemented the scanner.

5.3 Parser and AST

Files: *parser.mly, ast.ml*

The parser, written with OCamlYacc, takes in the tokens generated by the lexer and creates an Abstract Syntax Tree with the grammar defined in the parser and the datatypes defined in ast. The context-free grammar for the JavaLite syntax is defined in the parser through a set of production rules and precedence levels. If the source program is successfully parsed, it means the grammar is syntactically correct, and it will return the AST which is printed through pretty printing functions found in ast.ml. If there are violations, such as missing semicolons or missing arguments, the parser will throw errors.

Hongfei, Frances, and Mateo implemented the parser and AST.

5.4 Semantic Checking

Files: *semant.ml, sast.ml*

During semantic checking, the compiler traverses through the Abstract Syntax Tree created by the parser for type checking, and converts it to a semantically checked Abstract Syntax Tree. This step checks that types are consistent throughout each expression, variables are declared only once and in the proper scope, functions and classes are not redefined, among other things. This step is also important for built-in functions since LLVM requires types and argument types to be specified at compile time. If it passes semantic checking, it will return the SAST that can be printed through pretty printing functions found in sast.ml.

Hongfei and Frances implemented most of semantic checking, with a few additions from Mateo.

5.5 Code Generation

File: *codegen.ml*

The code generator takes in the semantically checked AST and constructs the LLVM code. The mappings between expressions and other features are accomplished with the OCaml LLVM module. On top of the int and double binary operations found in MicroC, we added string binary operations with the same modules with the help of LLVM build pointercast that creates a pointer to int instruction. The global print function is implemented in codegen through several helper functions that determine the type of the argument passed in. Further, it is at this step where modules are imported for the various built-in functions and converted to functions for Javalite in the call wrapper. In addition to functions, the code generation step also includes memory allocation for array and class type. More specifically, arrays are stored in the memory as pointers and each element is stored sequentially. For class, a named struct type would be created when a class is first declared and the values of all its fields are stored when the constructor gets called.

Hongfei and Frances implemented code generation.

5.6 C Libraries

Files: *stringfuncs.c*

Javalite provides several built-in functions for string and array manipulation that are reviewed in greater detail in section 3.5. Some of those functions were built with support from linked external C libraries.

Frances and Hongfei implemented the C libraries.

6 Test Plan

6.1 Representative Programs

GCD

```
int gcd(int a, int b) {
  while (a != b)
    if (a > b) a = a - b;
    else b = b - a;
  return a;
}

print(gcd(14,21));
print(gcd(8,36));
print(gcd(99,121));
```

Bubble Sort

```
int[] sortingArr = [52,14,72,5,66,7,12,31,9,3,54,41,53,12,61];
int i = 0;
int j = 0;
int tmpForSwap = 0;

int length = length(sortingArr);

for (i = 0 ; i < length-1 ; i = i + 1) {
for (j =0; j< length-i-1;j = j + 1){
  if (sortingArr[j] > sortingArr[j+1]){

    tmpForSwap = sortingArr[j];
    sortingArr[j] = sortingArr[j+1];
    sortingArr[j+1] = tmpForSwap;
  }
}
}

for (i = 0 ; i < length; i = i + 1) {
   print(sortingArr[i]);
}
```

Counting Sort

```
int[] countingSortArray = [0,0,0,0,0,0,0,0,0,0
            ,0,0,0,0,0,0,0,0,0,0
            ,0,0,0,0,0,0,0,0,0,0
            ,0,0,0,0,0,0,0,0,0,0
            ,0,0,0,0,0,0,0,0,0,0
            ];
int countingSortArrayLength = length(countingSortArray);
int[] sortingArr = [32,17,31,5,26,13,47,31,9,3,24,41,43,12,11];
int length = length(sortingArr);
int i = 0;
int j = 0;
```

```

for (i = 0 ; i < length ; i = i + 1) {
  countingSortArray[sortingArr[i]] = countingSortArray[sortingArr[i]] + 1;
}

for (i = 0 ; i < countingSortArrayLength ; i = i + 1) {
   if(countingSortArray[i] > 0){
    for(j=0;j<countingSortArray[i];j = j+1){
      print(i);
    }
   }
}
```

Class Declaration and Usage

```
class Person {
    string name;
    int age;
    string phrase;
}

void sayhi(class Person p) {
    string n = concat(p.name, " say:");
    print(n);
    string s = p.phrase;
    for (int i = 0; i < p.age; i = i + 1) {
        s = concat(s, p.phrase);
    }
    print(s);
}


class Person alice = Person("Alice", 3, "hey");
sayhi(alice);
```

The LLVM output for these sections will be in Section 6.5.

6.2 Testing Tools

The syntax and execution results can be tested by running *testall.sh* file, this will run all the test cases and check the execution result. Test cases are stored under the /test folder. Success test cases follow the format of *test-*.jl*, and Fail cases follow the format of *fail-*.jl*. Success cases have its corresponding *.out file as its expected output, while fail cases have its corresponding *.err file as its expected error.

To test the execution result of a single program, we can use the *BuildAndRun.sh* script. This will simply build the target source code file and show the execution result. E.g: If we run "*./BuildAndRun.sh SampleCode1.jl*" , it will build and execute *SampleCode1.jl*, and print out the execution result in the console.

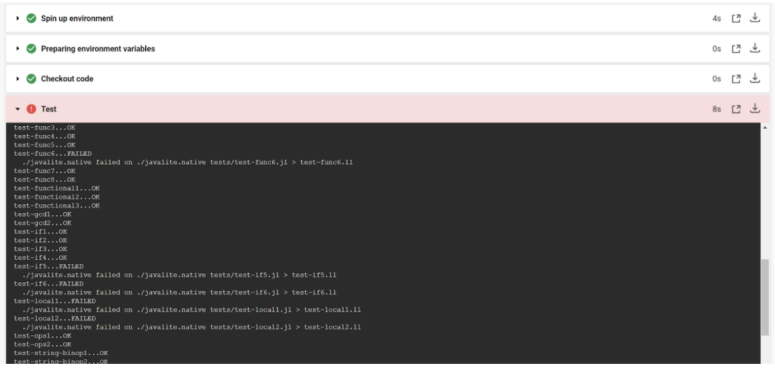
To benchmark the program performance, we can run the *PerformanceBench.sh*

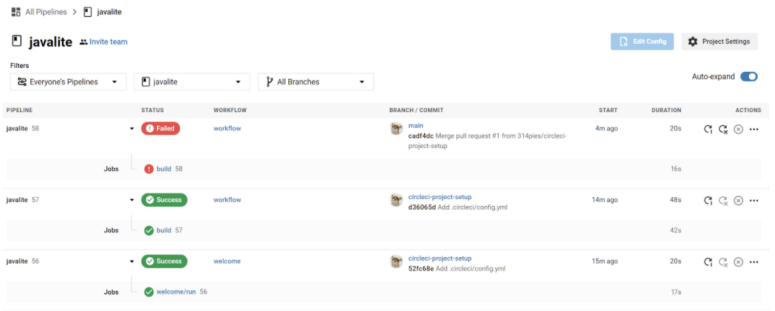
tool, this will benchmark all the performance tests under */test/performance* folder, and print out the test result to the console.

6.3 Automated CD/CI

For automated testing, we integrated CircleCI. Everytime a new commit is pushed, or a new pull request is created, CircleCI will spin up an container environment using *columbiasedwards/plt* docker image, and run the entire test suite to make sure everything passes. This automation process usually takes 20-60 seconds to complete.

The config.yml file for the automated testing process is located under the ./circleci directory in the source code repository.





6.4 Performance Benchmarking

The following is the performance benchmark result compared to Java. As the result shows, even though Javalite is easier to learn with a more Python-like syntax, the performance it has can be pretty close to Java. This may be improved even more by further optimizing the compiler.

To run Javalite performance benchmark, use *PerformanceBench.sh* in the root directory. All of the test cases used are under */test/performance* folder

	JavaLite	**Java**
Int-Arithmetic Test	2873 ms	1121 ms
Double-Arithmetic Test	4692 ms	3094 ms
If-Condition Test	2949 ms	1991 ms
Array-Access Test	4835 ms	3278 ms
Class-Field-Access Test	5268 ms	2289 ms

6.5 Target Language Program (LLVM)

These are the llvm outputs to some of our representative programs.

6.5.1 GCD

```
; ModuleID = 'JavaLite'
source_filename = "JavaLite"

@fmt = private unnamed_addr constant [4 x i8] c"%d\0A\00"
@fmt.1 = private unnamed_addr constant [4 x i8] c"%g\0A\00"
@fmt.2 = private unnamed_addr constant [4 x i8] c"%s\0A\00"
@fmt.3 = private unnamed_addr constant [4 x i8] c"%d\0A\00"
@fmt.4 = private unnamed_addr constant [4 x i8] c"%g\0A\00"
@fmt.5 = private unnamed_addr constant [4 x i8] c"%s\0A\00"

declare i32 @printf(i8*, ...)

declare i8* @reverse(i8*)

declare i8* @upper(i8*)

declare i8* @lower(i8*)

declare i8* @substring(i8*, i32, i32)

declare i32 @indexOf(i8*, i8*)

declare i32 @len(i8*)

declare i8* @concat(i8*, i8*)

declare i8* @to_string(i8*, ...)

declare i32 @length(i32*)

define i32 @main() {
entry:
  %gcd_result = call i32 @gcd(i32 14, i32 21)
  %printf = call i32 (i8*, ...) @printf(i8* getelementptr inbounds ([4 x i8], [4
    ↪ x i8]* @fmt, i32 0, i32 0), i32 %gcd_result)
  %gcd_result1 = call i32 @gcd(i32 8, i32 36)
  %printf2 = call i32 (i8*, ...) @printf(i8* getelementptr inbounds ([4 x i8], [4
    ↪  x i8]* @fmt, i32 0, i32 0), i32 %gcd_result1)
```

```
  %gcd_result3 = call i32 @gcd(i32 99, i32 121)
  %printf4 = call i32 (i8*, ...) @printf(i8* getelementptr inbounds ([4 x i8], [4
     ↪  x i8]* @fmt, i32 0, i32 0), i32 %gcd_result3)
  ret i32 0
}

define i32 @gcd(i32 %a, i32 %b) {
entry:
  %malloccall = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32*
     ↪  null, i32 1) to i32))
  %a1 = bitcast i8* %malloccall to i32*
  store i32 %a, i32* %a1
  %malloccall2 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32
     ↪ * null, i32 1) to i32))
  %b3 = bitcast i8* %malloccall2 to i32*
  store i32 %b, i32* %b3
  br label %while

while:                                            ; preds = %merge, %entry
  %a12 = load i32, i32* %a1
  %b13 = load i32, i32* %b3
  %tmp14 = icmp ne i32 %a12, %b13
  br i1 %tmp14, label %while_body, label %merge15

while_body:                                       ; preds = %while
  %a4 = load i32, i32* %a1
  %b5 = load i32, i32* %b3
  %tmp = icmp sgt i32 %a4, %b5
  br i1 %tmp, label %then, label %else

merge:                                            ; preds = %else, %then
  br label %while

then:                                             ; preds = %while_body
  %a6 = load i32, i32* %a1
  %b7 = load i32, i32* %b3
  %tmp8 = sub i32 %a6, %b7
  store i32 %tmp8, i32* %a1
  br label %merge

else:                                             ; preds = %while_body
  %b9 = load i32, i32* %b3
  %a10 = load i32, i32* %a1
  %tmp11 = sub i32 %b9, %a10
  store i32 %tmp11, i32* %b3
  br label %merge

merge15:                                          ; preds = %while
  %a16 = load i32, i32* %a1
  ret i32 %a16
}

declare noalias i8* @malloc(i32)
```

6.5.2 Bubble Sort

```
; ModuleID = 'JavaLite'
source_filename = "JavaLite"

@fmt = private unnamed_addr constant [4 x i8] c"%d\0A\00"
@fmt.1 = private unnamed_addr constant [4 x i8] c"%g\0A\00"
@fmt.2 = private unnamed_addr constant [4 x i8] c"%s\0A\00"

declare i32 @printf(i8*, ...)

declare i8* @reverse(i8*)

declare i8* @upper(i8*)

declare i8* @lower(i8*)

declare i8* @substring(i8*, i32, i32)

declare i32 @indexOf(i8*, i8*)

declare i32 @len(i8*)

declare i8* @concat(i8*, i8*)

declare i8* @to_string(i8*, ...)

declare i32 @length(i32*)

define i32 @main() {
entry:
  %malloccall = tail call i8* @malloc(i32 mul (i32 ptrtoint (i1** getelementptr (
     ↪ i1*, i1** null, i32 1) to i32), i32 15))
  %arr = bitcast i8* %malloccall to i32**
  %arrptr = bitcast i32** %arr to i32*
  %arrelt = getelementptr i32, i32* %arrptr, i32 0
  store i32 52, i32* %arrelt
  %arrelt1 = getelementptr i32, i32* %arrptr, i32 1
  store i32 14, i32* %arrelt1
  %arrelt2 = getelementptr i32, i32* %arrptr, i32 2
  store i32 72, i32* %arrelt2
  %arrelt3 = getelementptr i32, i32* %arrptr, i32 3
  store i32 5, i32* %arrelt3
  %arrelt4 = getelementptr i32, i32* %arrptr, i32 4
  store i32 66, i32* %arrelt4
  %arrelt5 = getelementptr i32, i32* %arrptr, i32 5
  store i32 7, i32* %arrelt5
  %arrelt6 = getelementptr i32, i32* %arrptr, i32 6
  store i32 12, i32* %arrelt6
  %arrelt7 = getelementptr i32, i32* %arrptr, i32 7
  store i32 31, i32* %arrelt7
  %arrelt8 = getelementptr i32, i32* %arrptr, i32 8
  store i32 9, i32* %arrelt8
  %arrelt9 = getelementptr i32, i32* %arrptr, i32 9
  store i32 3, i32* %arrelt9
  %arrelt10 = getelementptr i32, i32* %arrptr, i32 10
  store i32 54, i32* %arrelt10
  %arrelt11 = getelementptr i32, i32* %arrptr, i32 11
  store i32 41, i32* %arrelt11
```

```
  %arrelt12 = getelementptr i32, i32* %arrptr, i32 12
  store i32 53, i32* %arrelt12
  %arrelt13 = getelementptr i32, i32* %arrptr, i32 13
  store i32 12, i32* %arrelt13
  %arrelt14 = getelementptr i32, i32* %arrptr, i32 14
  store i32 61, i32* %arrelt14
  %malloccall15 = tail call i8* @malloc(i32 ptrtoint (i1** getelementptr (i1*, i1
     ↪ ** null, i32 1) to i32))
  %sortingArr = bitcast i8* %malloccall15 to i32**
  store i32* %arrptr, i32** %sortingArr
  %malloccall16 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32,
     ↪ i32* null, i32 1) to i32))
  %i = bitcast i8* %malloccall16 to i32*
  store i32 0, i32* %i
  %malloccall17 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32,
     ↪ i32* null, i32 1) to i32))
  %j = bitcast i8* %malloccall17 to i32*
  store i32 0, i32* %j
  %malloccall18 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32,
     ↪ i32* null, i32 1) to i32))
  %tmpForSwap = bitcast i8* %malloccall18 to i32*
  store i32 0, i32* %tmpForSwap
  %malloccall19 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32,
     ↪ i32* null, i32 1) to i32))
  %length = bitcast i8* %malloccall19 to i32*
  store i32 15, i32* %length
  store i32 0, i32* %i
  br label %while

while:                                            ; preds = %merge59, %entry
  %i62 = load i32, i32* %i
  %length63 = load i32, i32* %length
  %tmp64 = sub i32 %length63, 1
  %tmp65 = icmp slt i32 %i62, %tmp64
  br i1 %tmp65, label %while_body, label %merge66

while_body:                                       ; preds = %while
  store i32 0, i32* %j
  br label %while20

while20:                                          ; preds = %merge, %while_body
  %j53 = load i32, i32* %j
  %length54 = load i32, i32* %length
  %i55 = load i32, i32* %i
  %tmp56 = sub i32 %length54, %i55
  %tmp57 = sub i32 %tmp56, 1
  %tmp58 = icmp slt i32 %j53, %tmp57
  br i1 %tmp58, label %while_body21, label %merge59

while_body21:                                     ; preds = %while20
  %j22 = load i32, i32* %j
  %accpos = add i32 %j22, 0
  %sortingArr23 = load i32*, i32** %sortingArr
  %acceltptr = getelementptr i32, i32* %sortingArr23, i32 %accpos
  %accelt = load i32, i32* %acceltptr
  %j24 = load i32, i32* %j
  %tmp = add i32 %j24, 1
```

```
  %accpos25 = add i32 %tmp, 0
  %sortingArr26 = load i32*, i32** %sortingArr
  %acceltptr27 = getelementptr i32, i32* %sortingArr26, i32 %accpos25
  %accelt28 = load i32, i32* %acceltptr27
  %tmp29 = icmp sgt i32 %accelt, %accelt28
  br i1 %tmp29, label %then, label %else

merge:                                            ; preds = %else, %then
  %j51 = load i32, i32* %j
  %tmp52 = add i32 %j51, 1
  store i32 %tmp52, i32* %j
  br label %while20

then:                                             ; preds = %while_body21
  %j30 = load i32, i32* %j
  %accpos31 = add i32 %j30, 0
  %sortingArr32 = load i32*, i32** %sortingArr
  %acceltptr33 = getelementptr i32, i32* %sortingArr32, i32 %accpos31
  %accelt34 = load i32, i32* %acceltptr33
  store i32 %accelt34, i32* %tmpForSwap
  %j35 = load i32, i32* %j
  %accpos36 = add i32 %j35, 0
  %sortingArr37 = load i32*, i32** %sortingArr
  %j38 = load i32, i32* %j
  %tmp39 = add i32 %j38, 1
  %accpos40 = add i32 %tmp39, 0
  %sortingArr41 = load i32*, i32** %sortingArr
  %acceltptr42 = getelementptr i32, i32* %sortingArr41, i32 %accpos40
  %accelt43 = load i32, i32* %acceltptr42
  %arrelt44 = getelementptr i32, i32* %sortingArr37, i32 %accpos36
  store i32 %accelt43, i32* %arrelt44
  %j45 = load i32, i32* %j
  %tmp46 = add i32 %j45, 1
  %accpos47 = add i32 %tmp46, 0
  %sortingArr48 = load i32*, i32** %sortingArr
  %tmpForSwap49 = load i32, i32* %tmpForSwap
  %arrelt50 = getelementptr i32, i32* %sortingArr48, i32 %accpos47
  store i32 %tmpForSwap49, i32* %arrelt50
  br label %merge

else:                                             ; preds = %while_body21
  br label %merge

merge59:                                          ; preds = %while20
  %i60 = load i32, i32* %i
  %tmp61 = add i32 %i60, 1
  store i32 %tmp61, i32* %i
  br label %while

merge66:                                          ; preds = %while
  store i32 0, i32* %i
  br label %while67

while67:                                          ; preds = %while_body68, %
    ↪ merge66
  %i76 = load i32, i32* %i
  %length77 = load i32, i32* %length
```

```
  %tmp78 = icmp slt i32 %i76, %length77
  br i1 %tmp78, label %while_body68, label %merge79

while_body68:                                     ; preds = %while67
  %i69 = load i32, i32* %i
  %accpos70 = add i32 %i69, 0
  %sortingArr71 = load i32*, i32** %sortingArr
  %acceltptr72 = getelementptr i32, i32* %sortingArr71, i32 %accpos70
  %accelt73 = load i32, i32* %acceltptr72
  %printf = call i32 (i8*, ...) @printf(i8* getelementptr inbounds ([4 x i8], [4
     ↪ x i8]* @fmt, i32 0, i32 0), i32 %accelt73)
  %i74 = load i32, i32* %i
  %tmp75 = add i32 %i74, 1
  store i32 %tmp75, i32* %i
  br label %while67

merge79:                                          ; preds = %while67
  ret i32 0
}

declare noalias i8* @malloc(i32)
```

6.5.3 Class

```
; ModuleID = 'JavaLite'
source_filename = "JavaLite"

%Person = type { i8*, i32, i8* }

@fmt = private unnamed_addr constant [4 x i8] c"%d\0A\00"
@fmt.1 = private unnamed_addr constant [4 x i8] c"%g\0A\00"
@fmt.2 = private unnamed_addr constant [4 x i8] c"%s\0A\00"
@fmt.3 = private unnamed_addr constant [4 x i8] c"%d\0A\00"
@fmt.4 = private unnamed_addr constant [4 x i8] c"%g\0A\00"
@fmt.5 = private unnamed_addr constant [4 x i8] c"%s\0A\00"
@str = private unnamed_addr constant [4 x i8] c"hey\00"
@str.6 = private unnamed_addr constant [6 x i8] c"Alice\00"
@fmt.7 = private unnamed_addr constant [4 x i8] c"%d\0A\00"
@fmt.8 = private unnamed_addr constant [4 x i8] c"%g\0A\00"
@fmt.9 = private unnamed_addr constant [4 x i8] c"%s\0A\00"
@str.10 = private unnamed_addr constant [6 x i8] c" say:\00"

declare i32 @printf(i8*, ...)

declare i8* @reverse(i8*)

declare i8* @upper(i8*)

declare i8* @lower(i8*)

declare i8* @substring(i8*, i32, i32)

declare i32 @indexOf(i8*, i8*)

declare i32 @len(i8*)

```

```
declare i8* @concat(i8*, i8*)

declare i8* @to_string(i8*, ...)

declare i32 @length(i32*)

define %Person* @Person(i8* %name, i32 %age, i8* %phrase) {
entry:
  %malloccall = tail call i8* @malloc(i32 ptrtoint (i1** getelementptr (i1*, i1**
     ↪  null, i32 1) to i32))
  %name1 = bitcast i8* %malloccall to i8**
  store i8* %name, i8** %name1
  %malloccall2 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32
     ↪ * null, i32 1) to i32))
  %age3 = bitcast i8* %malloccall2 to i32*
  store i32 %age, i32* %age3
  %malloccall4 = tail call i8* @malloc(i32 ptrtoint (i1** getelementptr (i1*, i1
     ↪ ** null, i32 1) to i32))
  %phrase5 = bitcast i8* %malloccall4 to i8**
  store i8* %phrase, i8** %phrase5
  %malloccall6 = tail call i8* @malloc(i32 ptrtoint (%Person* getelementptr (%
     ↪ Person, %Person* null, i32 1) to i32))
  %constrObj = bitcast i8* %malloccall6 to %Person*
  %name7 = load i8*, i8** %name1
  %name8 = getelementptr inbounds %Person, %Person* %constrObj, i32 0, i32 0
  store i8* %name7, i8** %name8
  %age9 = load i32, i32* %age3
  %age10 = getelementptr inbounds %Person, %Person* %constrObj, i32 0, i32 1
  store i32 %age9, i32* %age10
  %phrase11 = load i8*, i8** %phrase5
  %phrase12 = getelementptr inbounds %Person, %Person* %constrObj, i32 0, i32 2
  store i8* %phrase11, i8** %phrase12
  %malloccall13 = tail call i8* @malloc(i32 ptrtoint (i1** getelementptr (i1*, i1
     ↪ ** null, i32 1) to i32))
  %obj = bitcast i8* %malloccall13 to %Person**
  store %Person* %constrObj, %Person** %obj
  %obj14 = load %Person*, %Person** %obj
  ret %Person* %obj14
}

define i32 @main() {
entry:
  %Person_result = call %Person* @Person(i8* getelementptr inbounds ([6 x i8], [6
     ↪  x i8]* @str.6, i32 0, i32 0), i32 3, i8* getelementptr inbounds ([4 x i8
     ↪ ], [4 x i8]* @str, i32 0, i32 0))
  %malloccall = tail call i8* @malloc(i32 ptrtoint (i1** getelementptr (i1*, i1**
     ↪  null, i32 1) to i32))
  %alice = bitcast i8* %malloccall to %Person**
  store %Person* %Person_result, %Person** %alice
  %alice1 = load %Person*, %Person** %alice
  call void @sayhi(%Person* %alice1)
  ret i32 0
}

define void @sayhi(%Person* %p) {
entry:
```

```
  %malloccall = tail call i8* @malloc(i32 ptrtoint (i1** getelementptr (i1*, i1**
     ↪  null, i32 1) to i32))
  %p1 = bitcast i8* %malloccall to %Person**
  store %Person* %p, %Person** %p1
  %p2 = load %Person*, %Person** %p1
  %objld = load %Person, %Person* %p2
  %name = extractvalue %Person %objld, 0
  %concat = call i8* @concat(i8* %name, i8* getelementptr inbounds ([6 x i8], [6
     ↪ x i8]* @str.10, i32 0, i32 0))
  %malloccall3 = tail call i8* @malloc(i32 ptrtoint (i1** getelementptr (i1*, i1
     ↪ ** null, i32 1) to i32))
  %n = bitcast i8* %malloccall3 to i8**
  store i8* %concat, i8** %n
  %n4 = load i8*, i8** %n
  %printf = call i32 (i8*, ...) @printf(i8* getelementptr inbounds ([4 x i8], [4
     ↪ x i8]* @fmt.9, i32 0, i32 0), i8* %n4)
  %p5 = load %Person*, %Person** %p1
  %objld6 = load %Person, %Person* %p5
  %phrase = extractvalue %Person %objld6, 2
  %malloccall7 = tail call i8* @malloc(i32 ptrtoint (i1** getelementptr (i1*, i1
     ↪ ** null, i32 1) to i32))
  %s = bitcast i8* %malloccall7 to i8**
  store i8* %phrase, i8** %s
  %malloccall8 = tail call i8* @malloc(i32 ptrtoint (i32* getelementptr (i32, i32
     ↪ * null, i32 1) to i32))
  %i = bitcast i8* %malloccall8 to i32*
  store i32 0, i32* %i
  br label %while

while:                                            ; preds = %while_body, %entry
  %i15 = load i32, i32* %i
  %p16 = load %Person*, %Person** %p1
  %objld17 = load %Person, %Person* %p16
  %age = extractvalue %Person %objld17, 1
  %tmp18 = icmp slt i32 %i15, %age
  br i1 %tmp18, label %while_body, label %merge

while_body:                                       ; preds = %while
  %p9 = load %Person*, %Person** %p1
  %objld10 = load %Person, %Person* %p9
  %phrase11 = extractvalue %Person %objld10, 2
  %s12 = load i8*, i8** %s
  %concat13 = call i8* @concat(i8* %s12, i8* %phrase11)
  store i8* %concat13, i8** %s
  %i14 = load i32, i32* %i
  %tmp = add i32 %i14, 1
  store i32 %tmp, i32* %i
  br label %while

merge:                                            ; preds = %while
  %s19 = load i8*, i8** %s
  %printf20 = call i32 (i8*, ...) @printf(i8* getelementptr inbounds ([4 x i8],
     ↪ [4 x i8]* @fmt.9, i32 0, i32 0), i8* %s19)
  ret void
}

declare noalias i8* @malloc(i32)
```

7 Lessons Learned

7.1 Frances Cao

This project requires full commitment and a ton of planning, which may be difficult for a few. OCaml is challenging at first, but after a bit of practice, it starts to click. One thing teams need to decide early on is whether they will be building their language with the MicroC architecture, which already compiles, or start fresh. Our team originally wanted to build our language from scratch, but realized it would be much better to work off of something that already works and build on top of it. Thus, another challenge is being able to understand the MicroC code, which is sizable, but once that happens, things start to get a little easier. Documentation is tough to understand at times, and there aren't many resources online, so reach out to the professor and T.As whenever you're stuck. Lastly, communication and collaboration is key, so make sure to voice issues early on. It's important that all team members hold their weight more or less, as building a compiler is not trivial.

7.2 Mateo Maturana

This project, while challenging, as also proven to be very rewarding when looking at the final project. At the beginning it does seem daunting, so my advice to future groups is to first start early and plan aggresively. The more you plan early on, the less burden you place on yourself and the rest of your team later one when it comes to making decisions. Additionally, you want to have a clear idea about how you want to build your language so work done during some of the earlier weeks is actually effective and makes it into the final project rather than making sideways or backwards progress and delaying the progress of your group. OCaml is not a simple language to learn as it is not very similar to languages many of us have used before such as Java, C, Python. Therefore, it is important that you spend time in the earlier weeks getting to know OCaml, reading documentation, and feeling comfortable programming in it. My final piece of advice is to study the MicroC code intensively, regardless of whether you plan to build your compiler on top of it or independent to it. By doing this, you are able to more easily construct your own compiler code since you have a much better idea of how the Codegen and Semant work and won't send as much time trying to debug these programs or just trying to figure out the syntax/logic.

7.3 Hongfei Chen

OCaml is an interesting programming language, but it could be challenging at first as it was new to almost everyone. However, when getting more familiar with OCaml as well as the LLVM library, programming becomes much more intuitive. One thing that I have learned, and I believe could be very helpful when implementing code generation part of the language, is starting with considering

memory allocation and how it would be done in C programming. Another aspect of the project was team collaboration- communication is crucial, especially when dealing with a project that has so many different components. If any issue was found, address it as early as possible and never have wishful thinking. Other than that, make good comments and notes, which would make it easier when trying to add in new features based on the old ones and this would also be great help when putting together the final report.

7.4 Ian Chen

Through this project, I learned how the compiler works and how a programming language is implemented. I now have a better understanding of how scanner, parser, ast, semant, sast, and codegen work, and how they can be modified to implement desired features for a new programming language. I also learned how new features can conflict with original codes and the importance of understanding the existing code before extending it. Before this project, I also didn't have much experience with MakeFile, shell script, and docker. But after this project, I now have a better idea of how to utilize them to construct an automatic workflow which could save a lot of time. This is also my first time setting up an automatic CD/CI using the docker image. Initially, I tried GitHub action, it took me a while to understand the docker command, yml configuration, and how to spin up a testing environment using the above combined. Unfortunately, even though I was pretty sure that the docker was already running, we were still not able to run through the tests using GitHub action, which is pretty frustrating. I then try to switch to CircleCI, this time is much faster as many CD/CI concepts I learned can already be applied. The testing environment spins-up correctly and we are able to run through all the tests with an automated workflow. Overall, through this project, I learned many valuable lessons and plenty of useful skills that can be used for future projects.

8 Appendix

8.1 scanner.mll

```
(* Template taken from MicroC *)

{ open Parser }

let digit = ['0' - '9']
let digits = digit+

rule token = parse
  [' ' '\t' '\r' '\n'] { token lexbuf } (* Whitespace *)
| "/*"     { comment lexbuf }           (* Multi-line Comments *)
| "//"     { singcomment lexbuf }       (* Single-line Comment *)
| '('      { LPAREN }
| ')'      { RPAREN }
| '{'      { LBRACE }
```

```
| '}'      { RBRACE }
| '['      { LBRACK }
| ']'      { RBRACK }
| ';'      { SEMI }
| ','      { COMMA }
| '+'      { PLUS }
| '-'      { MINUS }
| '*'      { TIMES }
| '/'      { DIVIDE }
| '='      { ASSIGN }
| "=="     { EQ }
| "!="     { NEQ }
| '<'      { LT }
| "<="     { LEQ }
| ">"      { GT }
| ">="     { GEQ }
| "&&"     { AND }
| "||"     { OR }
| "!"      { NOT }
| "if"     { IF }
| "else"   { ELSE }
| "for"    { FOR }
| "while"  { WHILE }
| "return" { RETURN }
| "int"    { INT }
| "bool"   { BOOL }
| "double"  { DOUBLE }
| "void"   { VOID }
| "string" { STRING }
| "true"   { BOOL_LIT(true)  }
| "false"  { BOOL_LIT(false) }
| "[]"     { ARRAY }
| "class"  { CLASS }
| "."      { DOT }
| "null"   { NULL }
(* | ":"      { EXTEND } *)
| digits as lxm { INT_LIT(int_of_string lxm) }
| digits '.'  digit* ( ['e' 'E'] ['+' '-']? digits )? as lxm { DOUBLE_LIT(lxm) }
| ['a'-'z' 'A'-'Z']['a'-'z' 'A'-'Z' '0'-'9' '_']*     as lxm { VARIABLE(lxm) }
| '"'       { STRING_LIT (stringbuf (Buffer.create 256) lexbuf) }
| eof { EOF }
| _ as char { raise (Failure("illegal character " ^ Char.escaped char)) }

and comment = parse
  "*/" { token lexbuf }
| _    { comment lexbuf }

and singcomment = parse
  "\n" { token lexbuf }
| _    { singcomment lexbuf }

and stringbuf buf = parse
  | '"'    { Buffer.contents buf }
  | _ as c { Buffer.add_char buf c; stringbuf buf lexbuf }
```

8.2 parser.mly

```
%{
open Ast
%}

%token LPAREN RPAREN LBRACE RBRACE LBRACK RBRACK
%token SEMI COMMA ASSIGN
%token NOT EQ NEQ LT LEQ GT GEQ AND OR
%token PLUS MINUS TIMES DIVIDE
%token RETURN IF ELSE FOR WHILE
%token INT BOOL DOUBLE VOID STRING
%token ARRAY
%token CLASS DOT
%token NULL

%token <int> INT_LIT
%token <bool> BOOL_LIT
%token <string> VARIABLE DOUBLE_LIT STRING_LIT
%token EOF

%start program
%type <Ast.program> program

%nonassoc NOELSE
%nonassoc ELSE
%right ASSIGN
%left OR
%left AND
%left EQ NEQ
%left LT GT LEQ GEQ
%left PLUS MINUS
%left TIMES DIVIDE
%right NOT

%%

program:
  decls EOF { $1 }

/* decls:
         { ([], [])               }
 | decls cdecl { (($2 :: fst $1), snd $1) }
 | decls fdecl { (fst $1, ($2 :: snd $1)) } */
decls:
    /* nothing */ { ([], [], []) }
  | decls stmt { let (stmt, cdecl, fdecl) = $1 in ($2::stmt, cdecl, fdecl) }
  | decls cdecl { let (stmt, cdecl, fdecl) = $1 in (stmt, $2::cdecl, fdecl) }
  | decls fdecl { let (stmt, cdecl, fdecl) = $1 in (stmt, cdecl, $2::fdecl) }


cdecl:
  CLASS VARIABLE LBRACE vdecl_list RBRACE
  { { cname = $2;
      fields = List.rev $4 } }


```

```
fdecl:
   typ VARIABLE LPAREN formals_opt RPAREN LBRACE stmt_list RBRACE
     { { typ = $1;
   fname = $2;
   formals = List.rev $4;
   locals = [];
   body = List.rev $7 } }

/*
sdecl:
    stmt_list     { $1 } */

formals_opt:
    /* nothing */ { [] }
  | formal_list   { $1 }

formal_list:
    typ VARIABLE                   { [($1,$2)]     }
  | formal_list COMMA typ VARIABLE { ($3,$4) :: $1 }

typ:
    INT   { Int   }
  | BOOL  { Bool  }
  | DOUBLE { Double }
  | VOID  { Void  }
  | STRING { String }
  | typ ARRAY { Arr($1, 0) }
  | CLASS VARIABLE { Object($2) }

vdecl_list:
    /* nothing */    { [] }
  | vdecl_list vdecl { $2 :: $1 }

vdecl:
   typ VARIABLE SEMI { ($1, $2) }

stmt_list:
    /* nothing */  { [] }
  | stmt_list stmt { $2 :: $1 }

stmt:
    expr SEMI                               { Expr $1               }
  | RETURN expr_opt SEMI                    { Return $2             }
  | LBRACE stmt_list RBRACE                 { Block(List.rev $2)    }
  | IF LPAREN expr RPAREN stmt %prec NOELSE { If($3, $5, Block([])) }
  | IF LPAREN expr RPAREN stmt ELSE stmt    { If($3, $5, $7)        }
  | FOR LPAREN expr_opt SEMI expr SEMI expr_opt RPAREN stmt
                                            { For($3, $5, $7, $9)   }
  | WHILE LPAREN expr RPAREN stmt           { While($3, $5)         }

expr_opt:
    /* nothing */ { Noexpr }
  | expr          { $1 }

expr:
    INT_LIT          { IntLit($1)             }
  | DOUBLE_LIT       { DoubleLit($1)          }
```

```
  | BOOL_LIT         { BoolLit($1)            }
  | STRING_LIT       { StrLit($1)             }
  | VARIABLE         { Var($1)                }
  | expr PLUS   expr { Binop($1, Add,   $3)   }
  | expr MINUS  expr { Binop($1, Sub,   $3)   }
  | expr TIMES  expr { Binop($1, Mult,  $3)   }
  | expr DIVIDE expr { Binop($1, Div,   $3)   }
  | expr EQ     expr { Binop($1, Equal, $3)   }
  | expr NEQ    expr { Binop($1, Neq,   $3)   }
  | expr LT     expr { Binop($1, Less,  $3)   }
  | expr LEQ    expr { Binop($1, Leq,   $3)   }
  | expr GT     expr { Binop($1, Greater, $3) }
  | expr GEQ    expr { Binop($1, Geq,   $3)   }
  | expr AND    expr { Binop($1, And,   $3)   }
  | expr OR     expr { Binop($1, Or,    $3)   }
  | MINUS expr %prec NOT { Unop(Neg, $2)      }
  | NOT expr          { Unop(Not, $2)          }
  | VARIABLE ASSIGN expr   { Assign($1, $3)         }
  | VARIABLE LPAREN args_opt RPAREN { Call($1, $3)  }
  | LPAREN expr RPAREN { $2                   }
  /* Arrays */
  | VARIABLE LBRACK expr RBRACK { ArrayAccess($1, $3) }
  | LBRACK args_list RBRACK { ArrayLit($2) }
  | VARIABLE LBRACK expr RBRACK ASSIGN expr { ArrAssign($1, $3, $6) }
  /* CALSS */
  | VARIABLE DOT VARIABLE { ObjAccess($1, $3) }
  | VARIABLE DOT VARIABLE ASSIGN expr   { ObjAssign($1, $3, $5) }
  /* VARIABLE DECLAREATION */
  | typ VARIABLE ASSIGN expr { DecAssn($1, $2, $4) }
  | NULL { NullPtr(Null) }


args_opt:
    /* nothing */ { [] }
  | args_list  { List.rev $1 }

args_list:
    expr                    { [$1] }
  | args_list COMMA expr { $3 :: $1 }
```

8.3 ast.ml

```
(* Abstract Syntax Tree and functions for printing it *)

type op = Add | Sub | Mult | Div | Equal | Neq | Less | Leq | Greater | Geq |
          And | Or

type uop = Neg | Not

type typ = Int | Bool | Double | Void | String | Arr of (typ * int) | Object of
    ↪ string | Any | Null

type bind = typ * string

type expr =
```

```
    IntLit of int
  | DoubleLit of string
  | BoolLit of bool
  | StrLit of string
  | Var of string
  | Binop of expr * op * expr
  | Unop of uop * expr
  | Assign of string * expr
  | Call of string * expr list
  | ArrayAccess of string * expr
  | ArrayLit of expr list
  | ArrAssign of string * expr * expr
  | ObjAccess of string * string
  | ObjAssign of string * string * expr
  | Construct of string * (string * expr) list
  | DecAssn of typ * string * expr
  | NullPtr of typ
  | Noexpr

type stmt =
    Block of stmt list
  | Expr of expr
  | Return of expr
  | If of expr * stmt * stmt
  | For of expr * expr * expr * stmt
  | While of expr * stmt

type func_decl = {
    typ : typ;
    fname : string;
    formals : bind list;
    locals : bind list;
    body : stmt list;
  }

type class_decl = {
  cname: string;
  fields: bind list;
}

type program = stmt list * class_decl list * func_decl list

(* Pretty-printing functions *)

let string_of_op = function
    Add -> "+"
  | Sub -> "-"
  | Mult -> "*"
  | Div -> "/"
  | Equal -> "=="
  | Neq -> "!="
  | Less -> "<"
  | Leq -> "<="
  | Greater -> ">"
  | Geq -> ">="
  | And -> "&&"
  | Or -> "||"
```

```
70
71  let string_of_uop = function
72      Neg -> "-"
73    | Not -> "!"
74
75  let rec string_of_typ = function
76      Int -> "int"
77    | Bool -> "bool"
78    | Double -> "double"
79    | Void -> "void"
80    | String -> "string"
81    | Any -> "any"
82    | Null -> "null"
83    | Arr(t, _) -> string_of_typ t ^ "[]"
84    | Object(s) -> "class " ^ s
85
86  let rec string_of_expr = function
87      IntLit(l) -> string_of_int l
88    | DoubleLit(l) -> l
89    | BoolLit(true) -> "true"
90    | BoolLit(false) -> "false"
91    | StrLit(s) -> "\"" ^ s ^ "\""
92    | Var(s) -> s
93    | Binop(e1, o, e2) ->
94        string_of_expr e1 ^ " " ^ string_of_op o ^ " " ^ string_of_expr e2
95    | Unop(o, e) -> string_of_uop o ^ string_of_expr e
96    | Assign(v, e) -> v ^ " = " ^ string_of_expr e
97    | Call(f, el) ->
98        f ^ "(" ^ String.concat ", " (List.map string_of_expr el) ^ ")"
99    | ArrayAccess (s, e) ->
100       s ^ "[" ^ string_of_expr e ^ "]"
101   | ArrayLit(e) -> "[" ^ String.concat "," (List.map string_of_expr (List.rev e))
        ↪  ^ "]"
102   | ArrAssign(s, e1, e2) -> s ^ "[" ^ string_of_expr e1 ^ "] = " ^ string_of_expr
        ↪  e2
103   | ObjAccess(s1, s2) -> s1 ^ "." ^ s2
104   | ObjAssign(s1, s2, e) -> s1 ^ "." ^ s2 ^ " = " ^ string_of_expr e
105   | Construct(s, _) -> "Constructor " ^ s
106   | DecAssn(t, s, e) -> string_of_typ t ^ " " ^ s ^ " = " ^ string_of_expr e
107   | NullPtr(t) -> string_of_typ t ^ " Null"
108   | Noexpr -> ""
109
110 let rec string_of_stmt = function
111     Block(stmts) ->
112       "{\n" ^ String.concat "" (List.map string_of_stmt stmts) ^ "}\n"
113   | Expr(expr) -> string_of_expr expr ^ ";\n";
114   | Return(expr) -> "return " ^ string_of_expr expr ^ ";\n";
115   | If(e, s, Block([])) -> "if (" ^ string_of_expr e ^ ")\n" ^ string_of_stmt s
116   | If(e, s1, s2) ->  "if (" ^ string_of_expr e ^ ")\n" ^
117       string_of_stmt s1 ^ "else\n" ^ string_of_stmt s2
118   | For(e1, e2, e3, s) ->
119       "for (" ^ string_of_expr e1  ^ " ; " ^ string_of_expr e2 ^ " ; " ^
120       string_of_expr e3  ^ ") " ^ string_of_stmt s
121   | While(e, s) -> "while (" ^ string_of_expr e ^ ") " ^ string_of_stmt s
122
123 let string_of_vdecl (t, id) = string_of_typ t ^ " " ^ id ^ ";\n"
124
```

```
let string_of_fdecl fdecl =
  string_of_typ fdecl.typ ^ " " ^
  fdecl.fname ^ "(" ^ String.concat ", " (List.map snd fdecl.formals) ^
  ")\n{\n" ^
  String.concat "" (List.map string_of_vdecl fdecl.locals) ^
  String.concat "" (List.map string_of_stmt fdecl.body) ^
  "}\n"

let string_of_cdecl cdecl =
  "class " ^ cdecl.cname ^ " {\n" ^
  String.concat "" (List.map string_of_vdecl cdecl.fields) ^
  "}\n"

let string_of_program (statements, classes, funcs) =
  (* String.concat "" (List.map string_of_vdecl vars) ^ "\n" ^ *)
  String.concat "\n" (List.map string_of_cdecl classes) ^ "\n" ^
  String.concat "\n" (List.map string_of_fdecl funcs) ^ "\n" ^
  String.concat "\n" (List.map string_of_stmt statements)
```

8.4 semant.ml

```
(* Semantic checking for the MicroC compiler *)

open Ast
open Sast

module StringMap = Map.Make(String);;

module HashtblString =
   struct
    type t = string
    let equal = ( = )
    let hash = Hashtbl.hash
   end;;

module StringHash = Hashtbl.Make(HashtblString);;

(* Semantic checking of the AST. Returns an SAST if successful,
   throws an exception if something is wrong.

   Check each global variable, then check each function *)

let check (statements, classes, functions) =

  (* Verify a list of bindings has no void types or duplicate names *)
  let check_binds (kind : string) (binds : bind list) =
    List.iter (function
        (Void, b) -> raise (Failure ("illegal void " ^ kind ^ " " ^ b))
      | _ -> ()) binds;
    let rec dups = function
        [] -> ()
      | ((_,n1) :: (_,n2) :: _) when n1 = n2 ->
    raise (Failure ("duplicate " ^ kind ^ " " ^ n1))
      | _ :: t -> dups t
    in dups (List.sort (fun (_,a) (_,b) -> compare a b) binds)
```

```
35  in
36
37  let check_bind tbl (t, n) =
38    if t = Void then raise (Failure ("illegal void " ^ n)) else
39    if (StringHash.mem tbl n) then raise (Failure ("duplicate " ^ n)) else
40      StringHash.add tbl n t; tbl
41  in
42
43  (**** Check classes ****)
44
45  (* Add class name to symbol table *)
46  let add_class map cd =
47    let dup_err = "duplicate class " ^ cd.cname in
48    let n = cd.cname in
49    match cd with
50      _ when StringMap.mem n map -> raise (Failure dup_err)
51    | _ -> StringMap.add n cd map
52  in
53
54  (* Collect all class names into one symbol table *)
55  let class_decls = List.fold_left add_class StringMap.empty classes
56  in
57
58  (* Return a function from our symbol table *)
59  let find_class s =
60    try StringMap.find s class_decls
61    with Not_found -> raise (Failure ("unrecognized class " ^ s))
62  in
63
64  let check_class cls =
65    (* Make sure no fields are void or duplicates *)
66    ignore (check_binds "field" cls.fields);
67    { scname = cls.cname;
68      sfields = cls.fields }
69  in
70
71  (* find the field bind in class *)
72  let find_field (cname, field) =
73    let cd = find_class cname in
74    let add_field m (t,n) = StringMap.add n (t,n) m in
75    let f_names = List.fold_left add_field StringMap.empty cd.fields in
76  try StringMap.find field f_names
77  with Not_found ->
78    raise (Failure ("unrecognized field " ^ field ^ " in class " ^ cname))
79  in
80
81  (**** Check functions ****)
82
83  (* Collect function declarations for built-in functions: no bodies *)
84  let built_in_print_decls =
85    let add_bind map name = StringMap.add name {
86      typ = Void;
87      fname = name;
88      formals = [(Any, "x")];
89      locals = []; body = [] } map
90    in List.fold_left add_bind StringMap.empty [ "print" ]
91  in
```

```
92
93  (* Array print support *)
94  (* let built_in_print_decls =
95    StringMap.add "to_string" {
96      typ = String;
97      fname = "to_string";
98      formals = [(Any, "x")];
99      locals = [];
100     body = [] } built_in_print_decls
101 in *)
102
103 let built_in_decls =
104   let add_str_func map (name, ty) = StringMap.add name {
105     typ = String;
106     fname = name;
107     formals = [(ty, "x")];
108     locals = []; body = [] } map
109   in List.fold_left add_str_func built_in_print_decls [ ("reverse", String);
110                                ("upper", String);
111                                ("lower", String) ]
112 in
113
114 let built_in_decls =
115   StringMap.add "substring" {
116     typ = String;
117     fname = "substring";
118     formals = [(String, "str"); (Int, "from"); (Int, "to")];
119     locals = [];
120     body = [] } built_in_decls
121  in
122
123  let built_in_decls =
124   StringMap.add "concat" {
125     typ = String;
126     fname = "concat";
127     formals = [(String, "str1"); (String, "str2")];
128     locals = [];
129     body = [] } built_in_decls
130  in
131
132  let built_in_decls =
133   StringMap.add "indexOf" {
134     typ = Int;
135     fname = "indexOf";
136     formals = [(String, "str"); (String, "find")];
137     locals = [];
138     body = [] } built_in_decls
139  in
140
141  let built_in_decls =
142   StringMap.add "len" {
143     typ = Int;
144     fname = "len";
145     formals = [(String, "str")];
146     locals = [];
147     body = [] } built_in_decls
148  in
```

```
149
150     let built_in_decls =
151      StringMap.add "length" {
152        typ = Int;
153        fname = "length";
154        formals = [(Any, "arr")];
155        locals = [];
156        body = [] } built_in_decls
157     in
158
159   (* declare main function with all statements *)
160   let main_decl =
161     {
162       typ = Int;
163       fname = "main";
164       formals = [];
165       locals = [];
166       body = List.rev statements }
167     in
168   let functions = main_decl :: functions in
169
170   (* Add function name to symbol table *)
171   let add_func map fd =
172     let built_in_err = "function " ^ fd.fname ^ " may not be defined"
173     and dup_err = "duplicate function " ^ fd.fname
174     and make_err er = raise (Failure er)
175     and n = fd.fname (* Name of the function *)
176     in match fd with (* No duplicate functions or redefinitions of built-ins *)
177          _ when StringMap.mem n built_in_decls -> make_err built_in_err
178        | _ when StringMap.mem n map -> make_err dup_err
179        | _ ->  StringMap.add n fd map
180   in
181
182   let form_constructor_body (cname, field_lst) =
183     let field_args = List.map (fun (_,n) -> (n,Var(n))) field_lst in
184     let return_body = [ Return (Var("obj")) ] in
185     let constructor = Expr (DecAssn(Object(cname), "obj", Construct(cname,
      ↪ field_args))) in
186     let body_rev = constructor :: return_body in
187     body_rev
188   in
189
190   let form_constructor cdecl =
191     let cn = cdecl.cname in
192     { typ = Object(cn);
193       fname = cn;
194       formals = cdecl.fields;
195       locals = [];
196       body = form_constructor_body (cn, cdecl.fields) }
197   in
198
199   let all_constructors = List.map (fun cd -> form_constructor cd) classes in
200
201   let all_functions = List.fold_left (fun l c -> c::l) functions all_constructors
      ↪  in
202
203   (* Collect all function names into one symbol table *)
```

```
204   let function_decls = List.fold_left add_func built_in_decls all_functions
205   in
206
207   (* Return a function from our symbol table *)
208   let find_func s =
209     try StringMap.find s function_decls
210     with Not_found -> raise (Failure ("unrecognized function " ^ s))
211   in
212
213   let rec check_arr (arrtyp, lv) =
214     (match arrtyp with
215       Arr(ty,_) ->  check_arr (ty, lv+1)
216     | _ -> (arrtyp, lv))
217   in
218
219   let _ = find_func "main" in (* Ensure "main" is defined *)
220
221   let check_function func =
222     (* Make sure no formals or locals are void or duplicates *)
223     check_binds "formal" func.formals;
224     check_binds "local" func.locals;
225     let tbl = StringHash.create 10 in
226     let formal_tbl = StringHash.create 5 in
227
228     let check_assign lvaluet rvaluet err =
229       if lvaluet = Any then rvaluet else
230         if lvaluet = rvaluet then lvaluet else
231           (match lvaluet with
232             Object(_) -> if rvaluet = Null then lvaluet else raise (Failure err)
233           | Arr _ -> (match rvaluet with
234                         Arr _ -> let r_arr = check_arr (rvaluet, 0) in
235                                  let l_arr = check_arr (lvaluet, 0) in
236                                    if r_arr = l_arr then rvaluet else raise (
    ↪ Failure err)
237                       | _ -> raise (Failure err))
238           | _ -> raise (Failure err))
239     in
240
241     (* Build local symbol table of variables for this function *)
242     let _ = List.fold_left check_bind
243       formal_tbl (func.formals @ func.locals )
244     in
245
246     (* Return a variable from our local symbol table *)
247     let type_of_identifier s =
248       try StringHash.find tbl s
249       with Not_found ->
250         try StringHash.find formal_tbl s
251         with Not_found -> raise (Failure ("undeclared identifier " ^ s))
252     in
253
254     (* check if of array type, return element type *)
255     let is_arr_ty (v, ty) = match ty with
256         Arr(t,_) ->
257           if t = Void then raise (Failure ("void type array " ^ v ^ " is not
    ↪ allowed"))
258           else t
```

```
259       | _ -> raise (Failure ("cannot access an element in variable " ^ v ^ " of
      ↪ type " ^ string_of_typ ty))
260     in
261
262     (* Return a semantically-checked expression, i.e., with a type *)
263     let rec expr e =
264       let check_assign_null e lt err =
265         let (rt, e') = expr e in
266         let ty = check_assign lt rt err in
267         if e' = (SNullPtr Null) then (ty, (SNullPtr ty)) else (ty, e')
268       in
269
270       (match e with
271         IntLit  l -> (Int, SIntLit l)
272       | DoubleLit l -> (Double, SDoubleLit l)
273       | BoolLit l  -> (Bool, SBoolLit l)
274       | StrLit s -> (String, SStrLit s)
275       | Noexpr     -> (Void, SNoexpr)
276       | NullPtr t -> (t, SNullPtr t)
277       | Var s       -> (type_of_identifier s, SVar s)
278       | Assign(var, e) as ex ->
279           let lt = type_of_identifier var in
280           let (rt, _) = expr e in
281           let err = "illegal assignment " ^ string_of_typ lt ^ " = " ^
282             string_of_typ rt ^ " in " ^ string_of_expr ex in
283           let (ty, e') = check_assign_null e lt err
284           (* update array size *)
285           in let _ = (match ty with
286             Arr _ -> StringHash.replace tbl var ty
287           | _ -> ignore 1)
288           in (ty, SAssign(var, (ty, e')))
289
290       | DecAssn(ty, var, e) as decassgn ->
291         ignore (check_bind tbl (ty, var));
292         let (rt, _) = expr e in
293         let err = "illegal assignment " ^ string_of_typ ty ^ " = " ^
294             string_of_typ rt ^ " in " ^ string_of_expr decassgn in
295         let (ty, e') = check_assign_null e ty err
296         (* update array size *)
297         in let _ = (match ty with
298           Arr _ -> StringHash.replace tbl var ty
299         | _ -> ignore 1)
300         in (ty, SDecAssn(ty, var, (ty, e')))
301
302       | Unop(op, e) as ex ->
303           let (t, e') = expr e in
304           let ty = match op with
305             Neg when t = Int || t = Double -> t
306           | Not when t = Bool -> Bool
307           | _ -> raise (Failure ("illegal unary operator " ^
308                                  string_of_uop op ^ string_of_typ t ^
309                                  " in " ^ string_of_expr ex))
310           in (ty, SUnop(op, (t, e')))
311       | Binop(e1, op, e2) as e ->
312           let (t1, e1') = expr e1
313           and (t2, e2') = expr e2 in
314           (* All binary operators require operands of the same type *)
```

```
315           let same = t1 = t2 in
316           (* Determine expression type based on operator and operand types *)
317           let ty = match op with
318             Add | Sub | Mult | Div when same && t1 = Int    -> Int
319           | Add | Sub | Mult | Div when same && t1 = Double -> Double
320           | Equal | Neq            when same               -> Bool
321           (* special case for null *)
322           | Equal | Neq when (t2 = Null) -> Bool
323           | Less | Leq | Greater | Geq
324                      when same && (t1 = Int || t1 = Double || t1 = String) ->
      ↪ Bool
325           | And | Or when same && t1 = Bool -> Bool
326           | Add when same && t1 = String -> String
327           | _ -> raise (
328         Failure ("illegal binary operator " ^
329                       string_of_typ t1 ^ " " ^ string_of_op op ^ " " ^
330                       string_of_typ t2 ^ " in " ^ string_of_expr e))
331         in (ty, SBinop((t1, e1'), op, (t2, e2')))
332       | Call(fname, args) as call ->
333         let fd = find_func fname in
334         let param_length = List.length fd.formals in
335         if List.length args != param_length then
336           raise (Failure ("expecting " ^ string_of_int param_length ^
337                           " arguments in " ^ string_of_expr call))
338         else let check_call (ft, n) e =
339           let (et, _) = expr e in
340           let err = "illegal argument found " ^ string_of_typ et ^
341             " expected " ^ string_of_typ ft ^ " in " ^ string_of_expr e
342           in let (ty,e') = check_assign_null e ft err
343             (* update array size *)
344             in let _ = (match ty with
345               Arr _ -> StringHash.replace formal_tbl n ty
346             | _ -> ignore 1) in (ty,e')
347         in
348         let args' = List.map2 check_call fd.formals args
349         in (fd.typ, SCall(fname, args'))
350
351       | ObjAccess(obj, f) as objaccess ->
352         let cname =
353           let obj_ty = type_of_identifier obj in
354           (match obj_ty with
355             Object(cn) -> cn
356           | _ -> raise (Failure (obj ^ " is not a class object in " ^
      ↪ string_of_expr objaccess)))
357         in
358         let (field_ty, _) = find_field (cname, f)
359         in (field_ty, SObjAccess(obj, cname, f))
360       | ObjAssign(obj, f, e) as objassign ->
361         let cname =
362           let obj_ty = type_of_identifier obj in
363           (match obj_ty with
364             Object(cn) -> cn
365           | _ -> raise (Failure (obj ^ " is not a class object in " ^
      ↪ string_of_expr objassign)))
366         in
367         let (field_ty, _) = find_field (cname, f) in
368         let (rt, _) = expr e in
```

```
369         let err = "illegal assignment " ^ string_of_typ field_ty ^ " = " ^
370             string_of_typ rt ^ " in " ^ string_of_expr objassign in
371         let (ty, e') = check_assign_null e field_ty err
372         in (ty, SObjAssign(obj, cname, f, (ty,e')))
373       | ArrayLit(el) as arraylit ->
374         (* check if types of expr are consistent *)
375           let ty_inconsistent_err = "inconsistent types in array " ^
    ↪ string_of_expr arraylit in
376           let fst_e = List.hd el in
377           let (fst_ty, _) = expr fst_e in
378           let (arr_ty_len, arr_ty_e) = List.fold_left (fun (t, l) e ->
379             let (et, e') = expr e in
380             let is_arr = (
381               match et with
382                 Arr _ -> true
383               | _ -> false) in
384             if ((et = fst_ty) || is_arr) then (t+1, (et, e')::l) else (t, (et, e
    ↪ ')::l)) (0,[]) el
385           in if arr_ty_len != List.length el
386             then raise (Failure ty_inconsistent_err)
387           (* determine arr type *)
388           else let arr_ty = Arr(fst_ty, arr_ty_len)
389         in (arr_ty, SArrayLit(arr_ty_e))
390       | Construct(cname, args) ->
391         let args' = List.map (fun (s,e) -> (s, (expr e))) args
392         in (Object(cname), SConstruct(cname, args'))
393       | ArrayAccess(v, e) as arrayacess ->
394           (* check if type of e is an int *)
395           let (t, e') = expr e in
396           if t != Int then raise (Failure (string_of_expr e ^ " is not of int
    ↪ type in " ^ string_of_expr arrayacess))
397           else
398           (* check if variable is array type *)
399           let v_ty = type_of_identifier v in
400           let e_ty = is_arr_ty (v, v_ty)
401         in (e_ty, SArrayAccess(v, (t, e')))
402       | ArrAssign(v, e1, e2) as arrassign ->
403         (* check if type of e is an int *)
404         let (t, e1') = expr e1 in
405         if t != Int then raise (Failure (string_of_expr e1 ^ " is not of int type
    ↪  in " ^ string_of_expr arrassign))
406         else
407         (* check if variable is array type *)
408         let v_ty = type_of_identifier v in
409         let e_ty = is_arr_ty (v, v_ty) in
410         let (rt, _) = expr e2 in
411         let err = "illegal assignment " ^ string_of_typ e_ty ^ " = " ^
412             string_of_typ rt ^ " in " ^ string_of_expr arrassign in
413         let (ty, e2') = check_assign_null e2 e_ty err
414         in (ty, SArrAssign(v, (t,e1'), (ty,e2')))
415       )
416     in
417
418     let check_bool_expr e =
419       let (t', e') = expr e
420       and err = "expected Boolean expression in " ^ string_of_expr e
421       in if t' != Bool then raise (Failure err) else (t', e')
```

```
422     in
423
424     (* Return a semantically-checked statement i.e. containing sexprs *)
425     let rec check_stmt = function
426         Expr e -> SExpr (expr e)
427       | If(p, b1, b2) -> SIf(check_bool_expr p, check_stmt b1, check_stmt b2)
428       | For(e1, e2, e3, st) ->
429         let e1' = expr e1 in
430         SFor(e1', check_bool_expr e2, expr e3, check_stmt st)
431       | While(p, s) -> SWhile(check_bool_expr p, check_stmt s)
432       | Return e -> let (t, e') = expr e in
433         if "main" = func.fname then
434         raise (Failure("expected no return statement in outer body")) else
435         let err = "return gives " ^ string_of_typ t ^ " expected " ^
436           string_of_typ func.typ ^ " in " ^ string_of_expr e in
437         if t = func.typ then SReturn (t, e') else
438           (match t with
439             Arr _ -> (match func.typ with
440                         Arr _ -> let r_arr = check_arr (func.typ, 0) in
441                                  let l_arr = check_arr (t, 0) in
442                                    if r_arr = l_arr then SReturn (func.typ, e')
    ↪ else raise (Failure err)
443                       | _ -> raise (Failure err))
444           | _ -> raise (Failure err))
445
446       (* A block is correct if each statement is correct and nothing
447          follows any Return statement.  Nested blocks are flattened. *)
448       | Block sl ->
449           let rec check_stmt_list = function
450             [Return _ as s] -> [check_stmt s]
451             | Return _ :: _   -> raise (Failure "nothing may follow a return")
452             | Block sl :: ss  -> check_stmt_list (sl @ ss) (* Flatten blocks *)
453             | s :: ss         -> let c = check_stmt s in
454                                      c :: check_stmt_list ss
455             | []              -> []
456           in SBlock(check_stmt_list sl)
457
458     in (* body of check_function *)
459     { styp = func.typ;
460       sfname = func.fname;
461       sformals = func.formals;
462       slocals  = func.locals;
463       sbody = match check_stmt (Block func.body) with
464   SBlock(sl) -> sl
465       | _ -> raise (Failure ("internal error: block didn't become a block?"))
466     }
467   in (List.map check_class classes, List.map check_function all_functions)
```

8.5 sast.ml

```
1  (* Semantically-checked Abstract Syntax Tree and functions for printing it *)
2
3  open Ast
4
5  type sexpr = typ * sx
```

```
and sx =
    SIntLit of int
  | SDoubleLit of string
  | SBoolLit of bool
  | SStrLit of string
  | SVar of string
  | SBinop of sexpr * op * sexpr
  | SUnop of uop * sexpr
  | SAssign of string * sexpr
  | SCall of string * sexpr list
  | SArrayAccess of string * sexpr
  | SArrayLit of sexpr list
  | SArrAssign of string * sexpr * sexpr
  | SObjAccess of string * string * string
  | SObjAssign of string * string * string * sexpr
  (* cname * (field_name * field_value) list *)
  | SConstruct of string * (string * sexpr) list
  | SDecAssn of typ * string * sexpr
  | SNullPtr of typ
  | SNoexpr

type sstmt =
    SBlock of sstmt list
  | SExpr of sexpr
  | SReturn of sexpr
  | SIf of sexpr * sstmt * sstmt
  | SFor of sexpr * sexpr * sexpr * sstmt
  | SWhile of sexpr * sstmt

type sfunc_decl = {
    styp : typ;
    sfname : string;
    sformals : bind list;
    slocals : bind list;
    sbody : sstmt list;
  }

type sclass_decl = {
  scname : string;
  sfields : bind list;
}

type sprogram = sclass_decl list * sfunc_decl list

(* Pretty-printing functions *)

let rec string_of_sexpr (t, e) =
  "(" ^ string_of_typ t ^ " : " ^ (match e with
    SIntLit(l) -> string_of_int l
  | SBoolLit(true) -> "true"
  | SBoolLit(false) -> "false"
  | SDoubleLit(l) -> l
  | SStrLit(s) -> "\"" ^ s ^ "\""
  | SVar(s) -> s
  | SBinop(e1, o, e2) ->
      string_of_sexpr e1 ^ " " ^ string_of_op o ^ " " ^ string_of_sexpr e2
  | SUnop(o, e) -> string_of_uop o ^ string_of_sexpr e
```

```
  | SAssign(v, e) -> v ^ " = " ^ string_of_sexpr e
  | SCall(f, el) ->
      f ^ "(" ^ String.concat ", " (List.map string_of_sexpr el) ^ ")"
  | SArrayAccess(s, e) ->
      s ^ "[" ^ string_of_sexpr e ^ "]"
  | SArrayLit(e) -> "[" ^ String.concat "," (List.map string_of_sexpr (List.rev e
    ↪ )) ^ "]"
  | SArrAssign(s, e1, e2) -> s ^ "[" ^ string_of_sexpr e1 ^ "] = " ^
    ↪ string_of_sexpr e2
  | SObjAccess(s1, c, s2) -> s1 ^ "(class " ^ c ^ ")." ^ s2
  | SObjAssign(s1, c, s2, e) -> s1 ^ "(class " ^ c ^ ")." ^ s2 ^ " = " ^
    ↪ string_of_sexpr e
  | SConstruct(s, _) -> "Constructor " ^ s
  | SDecAssn(t, s, e) -> string_of_typ t ^ " " ^ s ^ " = " ^ string_of_sexpr e
  | SNullPtr(t) ->  string_of_typ t ^ " Null"
  | SNoexpr -> ""
        ) ^ ")"

let print_sstring (_, e) = match e with
    SStrLit(s) -> s
  | _ -> raise (Failure "error: only string type allowed")

let compare_sstring ((_, e1),(_,e2)) =
    if SStrLit(e1) = SStrLit(e2) then true
    else false

let rec string_of_sstmt = function
    SBlock(stmts) ->
      "{\n" ^ String.concat "" (List.map string_of_sstmt stmts) ^ "}\n"
  | SExpr(expr) -> string_of_sexpr expr ^ ";\n";
  | SReturn(expr) -> "return " ^ string_of_sexpr expr ^ ";\n";
  | SIf(e, s, SBlock([])) ->
      "if (" ^ string_of_sexpr e ^ ")\n" ^ string_of_sstmt s
  | SIf(e, s1, s2) ->  "if (" ^ string_of_sexpr e ^ ")\n" ^
      string_of_sstmt s1 ^ "else\n" ^ string_of_sstmt s2
  | SFor(e1, e2, e3, s) ->
      "for (" ^ string_of_sexpr e1  ^ " ; " ^ string_of_sexpr e2 ^ " ; " ^
      string_of_sexpr e3  ^ ") " ^ string_of_sstmt s
  | SWhile(e, s) -> "while (" ^ string_of_sexpr e ^ ") " ^ string_of_sstmt s

let string_of_sfdecl fdecl =
  string_of_typ fdecl.styp ^ " " ^
  fdecl.sfname ^ "(" ^ String.concat ", " (List.map snd fdecl.sformals) ^
  ")\n{\n" ^
  String.concat "" (List.map string_of_vdecl fdecl.slocals) ^
  String.concat "" (List.map string_of_sstmt fdecl.sbody) ^
  "}\n"

let string_of_scdecl cdecl =
  "class " ^ cdecl.scname ^ " {\n" ^
  String.concat "" (List.map string_of_vdecl cdecl.sfields) ^
  "}\n"

let string_of_sprogram (classes, funcs) =
  (* String.concat "" (List.map string_of_vdecl vars) ^ "\n" ^ *)
  String.concat "\n" (List.map string_of_scdecl classes) ^ "\n" ^
  String.concat "\n" (List.map string_of_sfdecl funcs)
```

8.6 codegen.ml

```
(* Code generation: translate takes a semantically checked AST and
produces LLVM IR

LLVM tutorial: Make sure to read the OCaml version of the tutorial

http://llvm.org/docs/tutorial/index.html

Detailed documentation on the OCaml LLVM library:

http://llvm.moe/
http://llvm.moe/ocaml/

*)

module L = Llvm
module A = Ast
open Sast

module StringMap = Map.Make(String);;

module HashtblString =
   struct
    type t = string
    let equal = ( = )
    let hash = Hashtbl.hash
   end;;

module StringHash = Hashtbl.Make(HashtblString);;

(* translate : Sast.program -> Llvm.module *)
let translate (classes, functions) =
  let context    = L.global_context () in

  (* Create the LLVM compilation module into which
     we will generate code *)
  let the_module = L.create_module context "JavaLite" in

  (* Get types from the context *)
  let i32_t      = L.i32_type    context
  and i8_t       = L.i8_type     context
  and i1_t       = L.i1_type     context
  and double_t   = L.double_type context
  and void_t     = L.void_type   context in

  (* string type *)
  let string_t   = L.pointer_type i8_t in

  let add_class_typ map cd =
    let cname = cd.scname in
    let typ_lst = List.map (fun (t,_) -> t) cd.sfields in
    StringMap.add cname typ_lst map
  in
  (* store all class name and corresponding field type list *)
  let class_types = List.fold_left add_class_typ StringMap.empty classes
  in
```

```
(* get the list of field types for a class *)
let find_field_typs cname =
  try StringMap.find cname class_types
  with Not_found -> raise (Failure ("find_field_typs not found " ^ cname))
in

let add_field_name map cd =
  let cname = cd.scname in
  let field_lst = List.map (fun (_,n) -> n) cd.sfields in
  StringMap.add cname field_lst map
in

let class_fields = List.fold_left add_field_name StringMap.empty classes
in

let add_cd m cd = StringMap.add cd.scname cd.sfields m in
let class_ty_fields = List.fold_left add_cd StringMap.empty classes
in

let get_field_ind (cname, fname) =
  let f_lst =
    try StringMap.find cname class_fields
    with Not_found -> raise (Failure ("get_field_ind not found " ^ cname))
   in
  let f_lst_ind = List.mapi (fun i n -> (n, i)) f_lst in
  snd (List.hd (List.filter (fun (n, _) -> n = fname) f_lst_ind))
in

let class_tbl = StringHash.create 10 in
let find_struct cls =
  try StringHash.find class_tbl cls
  with Not_found ->
    let cls_struct = L.named_struct_type context cls in
    StringHash.add class_tbl cls cls_struct; cls_struct

in
(* Return the LLVM type for a MicroC type *)
let rec ltype_of_typ = function
    A.Int    -> i32_t
  | A.Bool   -> i1_t
  | A.Double -> double_t
  | A.Void   -> void_t
  | A.String -> string_t
  | A.Arr(ty,_) -> L.pointer_type (ltype_of_typ ty)
  | A.Object(cls) -> L.pointer_type (find_struct cls)
  | _        -> raise (Failure "Unmatched LLVM type in ltype_of_typ")
in

(* Import modules for our built-in functions and print *)
let printf_t : L.lltype =
    L.var_arg_function_type i32_t [| string_t |] in
let printf_func : L.llvalue =
    L.declare_function "printf" printf_t the_module in
let reversestring_t : L.lltype =
    L.function_type string_t [| string_t |] in
let reversestring_func : L.llvalue =
    L.declare_function "reverse" reversestring_t the_module in
```

```
113
114   let stringupper_t : L.lltype =
115       L.function_type string_t [| string_t |] in
116   let stringupper_func : L.llvalue =
117       L.declare_function "upper" stringupper_t the_module in
118
119   let stringlower_t : L.lltype =
120       L.function_type string_t [| string_t |] in
121   let stringlower_func : L.llvalue =
122       L.declare_function "lower" stringlower_t the_module in
123
124   let stringsubstring_t : L.lltype =
125       L.function_type string_t [| string_t ; i32_t ; i32_t |] in
126   let stringsubstring_func : L.llvalue =
127       L.declare_function "substring" stringsubstring_t the_module in
128
129   let stringindexof_t : L.lltype =
130       L.function_type i32_t [| string_t ; string_t |] in
131   let stringindexof_func : L.llvalue =
132       L.declare_function "indexOf" stringindexof_t the_module in
133
134   let stringlen_t : L.lltype =
135       L.function_type i32_t [| string_t |] in
136   let stringlen_func : L.llvalue =
137       L.declare_function "len" stringlen_t the_module in
138
139   let stringconcat_t : L.lltype =
140       L.function_type string_t [| string_t ; string_t |] in
141   let stringconcat_func : L.llvalue =
142       L.declare_function "concat" stringconcat_t the_module in
143
144   (* Array functions *)
145   let to_string_t : L.lltype =
146       L.var_arg_function_type string_t [| string_t |] in
147   let to_string : L.llvalue =
148       L.declare_function "to_string" to_string_t the_module in
149
150   (* let arraylen_t : L.lltype =
151       L.function_type i32_t [| L.pointer_type i32_t |] in
152   let arraylen_func : L.llvalue =
153       L.declare_function "length" arraylen_t the_module in *)
154
155   (* Define each function (arguments and return type) so we can
156      call it even before we've created its body *)
157   let function_decls : (L.llvalue * sfunc_decl) StringMap.t =
158     let function_decl m fdecl =
159       let name = fdecl.sfname
160       and formal_types =
161   Array.of_list (List.map (fun (t,_) -> ltype_of_typ t) fdecl.sformals)
162       in let ftype = L.function_type (ltype_of_typ fdecl.styp) formal_types in
163       StringMap.add name (L.define_function name ftype the_module, fdecl) m in
164     List.fold_left function_decl StringMap.empty functions in
165
166   (* Fill in the body of the given function *)
167   let build_function_body fdecl =
168     let (the_function, _) =
169       try StringMap.find fdecl.sfname function_decls
```

```
      with Not_found -> raise (Failure ("build_function_body not found " ^ fdecl.
    ↪ sfname))
  in
  let builder = L.builder_at_end context (L.entry_block the_function) in

  let int_format_str = L.build_global_stringptr "%d\n" "fmt" builder
  and float_format_str = L.build_global_stringptr "%g\n" "fmt" builder
  and str_format_str = L.build_global_stringptr "%s\n" "fmt" builder
  in

  (* Find the type of the input *)
  let find_str_typ = function
      A.Int     -> int_format_str
    | A.Bool    -> int_format_str
    | A.Double  -> float_format_str
    | A.String  -> str_format_str
    | _         -> raise (Failure "Invalid type")
  in

  (* Construct a hash table for function formals and locals
     add all the formals first *)
  let tbl = StringHash.create 10 in
  let formal_tbl = StringHash.create 5 in
    let add_formal tbl (t, n) p =
      L.set_value_name n p;
      let local = L.build_alloca (ltype_of_typ t) n builder in
          ignore (L.build_store p local builder);
      StringHash.add tbl n local; tbl in
    let _ = List.fold_left2 add_formal formal_tbl fdecl.sformals
      (Array.to_list (L.params the_function)) in


  (* Return the value for a variable or formal argument.
     Check local names first, then global names *)
  let lookup n = try StringHash.find tbl n
                 with Not_found ->
                   try StringHash.find formal_tbl n
                   with Not_found -> raise (Failure ("variable " ^ n ^ " not
    ↪ found in lookup"))
  in

  let i32_t_pt = L.string_of_lltype i32_t in
  let i1_t_pt = L.string_of_lltype i1_t in
  let double_t_pt = L.string_of_lltype double_t in
  let string_t_pt = L.string_of_lltype string_t in

  let match_typ t =
    let t_pt = L.string_of_lltype t in
    if t_pt = string_t_pt then A.String else
      if t_pt = double_t_pt then A.Double else
        if t_pt = i32_t_pt then A.Int else
          if t_pt = i1_t_pt then A.Int else
            raise (Failure ("cannot match type: " ^ t_pt)) in

  let is_arith = function
      A.Add     -> true
    | A.Sub     -> true
```

```
225       | A.Mult    -> true
226       | A.Div     -> true
227       | _         -> false in
228
229     let rec find_type = function
230       | SIntLit _     -> A.Int
231       | SBoolLit _     -> A.Int
232       | SDoubleLit _   -> A.Double
233       | SStrLit _      -> A.String
234       | SBinop((_, e_x), op, _) -> if (is_arith op) = true then find_type e_x
235                                    else A.Int
236       | SUnop(_, (_, e_x)) -> find_type e_x
237       | SNoexpr        -> raise (Failure "Unmatched NoExpr")
238       | SNullPtr _ -> A.String
239       | SCall(f, _) -> let (_, fdecl) = StringMap.find f function_decls in
240                            (match fdecl.styp with
241                              A.Void -> raise (Failure "Cannot print void")
242                              | _     -> fdecl.styp)
243       | SVar v        -> match_typ (L.element_type (L.type_of (lookup v)))
244       | SAssign(v, _)        -> find_type (SVar(v))
245       | SArrayAccess(s, _)   -> match_typ (L.element_type (L.element_type (L.
      ↪ type_of (lookup s))))
246       | SObjAccess(_, c, f) ->
247         let f_lst = StringMap.find c class_ty_fields in
248         fst (List.hd (List.filter (fun (_,n) -> n = f) f_lst))
249         (* todo *)
250       | SArrayLit (el)    ->  let (_, e_fst) = (List.nth el 0) in
251           A.Arr (find_type e_fst, List.length el)
252       | SConstruct _ -> raise (Failure "SConstruct cannot be printed")
253       | SArrAssign _ -> raise (Failure "SArrAssign cannot be printed")
254       | SObjAssign _ -> raise (Failure "SObjAssign cannot be printed")
255       | SDecAssn _ -> raise (Failure "SDecAssn not implemented")
256     in
257
258     (* Construct code for an expression; return its value *)
259     let rec expr builder ((_, e) : sexpr) = match e with
260         SIntLit i  -> L.const_int i32_t i
261       | SBoolLit b  -> L.const_int i1_t (if b then 1 else 0)
262       | SDoubleLit l -> L.const_float_of_string double_t l
263       | SStrLit s -> L.build_global_stringptr s "str" builder
264       | SArrayLit arr   ->
265         (* arr: sexpr list = typ * sx list *)
266         (* let len = L.const_int i32_t ((List.length arr) + 1) in *)
267         let len = L.const_int i32_t (List.length arr) in
268         let size = L.const_int i32_t ((List.length arr) + 1) in
269         let (fst_t, _) = List.hd arr in
270         let ty = ltype_of_typ (A.Arr(fst_t, (List.length arr))) in
271         (* allocate memory for array *)
272         let arr_alloca = L.build_array_alloca ty size "arr" builder in
273         (* bitcast *)
274         let arr_ptr = L.build_pointercast arr_alloca ty "arrptr" builder in
275         (* store all elements *)
276         let elts = List.map (expr builder) arr in
277         let store_elt ind elt =
278           let pos = L.const_int i32_t (ind) in
279             let elt_ptr = L.build_gep arr_ptr [| pos |] "arrelt" builder in
280           ignore(L.build_store elt elt_ptr builder)
```

```
281         in List.iteri store_elt elts;
282         let elt_ptr = L.build_gep arr_ptr [| len |] "arrlast" builder in
283         let null_elt = L.const_null (L.element_type ty) in
284         ignore(L.build_store null_elt elt_ptr builder);
285         arr_ptr
286       | SConstruct (cname, args) ->
287         (* let obj_ty = ltype_of_typ (A.Object(cname)) in  *)
288         let obj_ty = find_struct cname in
289         let fields' = List.map (fun ty -> ltype_of_typ ty) (find_field_typs cname
    ↪ ) in
290         let _ = L.struct_set_body obj_ty (Array.of_list fields') false in
291         let obj_alloca = L.build_malloc obj_ty "constrObj" builder in
292         (* bitcast *)
293         let obj_ptr = L.build_pointercast obj_alloca (L.pointer_type obj_ty) "
    ↪ constrPtr" builder in
294         (* store all fields *)
295         let store_field (fname, e) =
296           let f_val = expr builder e in
297           let ind = get_field_ind (cname, fname) in
298           let f_ptr = L.build_struct_gep obj_ptr ind fname builder in
299             ignore(L.build_store f_val f_ptr builder)
300         in List.iter store_field args; obj_ptr
301       | SArrayAccess (s, e)  ->
302         let ind = expr builder e in
303         let (ty, _) = e in
304         (* increment index by one to get actual ptr position *)
305         let pos = L.build_add ind (L.const_int i32_t 0) "accpos" builder in
306         let arr = expr builder (ty, (SVar s)) in
307         let elt = L.build_gep arr [| pos |] "acceltptr" builder in
308         L.build_load elt "accelt" builder
309       | SObjAccess (s, cname, f) ->
310         let obj = expr builder (A.String, (SVar s)) in
311         let obj_ptr = L.build_load obj "objld" builder in
312         let ind = get_field_ind (cname, f) in
313         L.build_extractvalue obj_ptr ind f builder
314       | SArrAssign (s, e1, e2) ->
315         let ind = expr builder e1 in
316         let (ty, _) = e1 in
317         (* increment index by one to get actual ptr position *)
318         let pos = L.build_add ind (L.const_int i32_t 0) "accpos" builder in
319         let arr = expr builder (ty, (SVar s)) in
320         let new_val = expr builder e2 in
321         let elt_ptr = L.build_gep arr [| pos |] "arrelt" builder in
322         L.build_store new_val elt_ptr builder
323       | SObjAssign (s, cname, f, e) ->
324         let obj = expr builder (A.String, (SVar s)) in
325         let f_val = expr builder e in
326         let ind = get_field_ind (cname, f) in
327         let f_ptr = L.build_struct_gep obj ind f builder in
328         L.build_store f_val f_ptr builder
329       | SNoexpr     -> L.const_int i32_t 0
330       | SNullPtr t -> L.const_pointer_null (ltype_of_typ t)
331       | SVar s       -> L.build_load (lookup s) s builder
332       | SAssign (s, e) -> let e' = expr builder e in
333                           ignore(L.build_store e' (lookup s) builder); e'
334       | SDecAssn (t, s, e) ->
335         let e' = expr builder e in
```

```
336         let var = L.build_alloca (ltype_of_typ t) s builder in
337             ignore (L.build_store e' var builder);
338         StringHash.add tbl s var; e'
339       | SBinop ((A.Double,_ ) as e1, op, e2) ->
340     let e1' = expr builder e1
341     and e2' = expr builder e2 in
342     (match op with
343       A.Add     -> L.build_fadd
344     | A.Sub     -> L.build_fsub
345     | A.Mult    -> L.build_fmul
346     | A.Div     -> L.build_fdiv
347     | A.Equal   -> L.build_fcmp L.Fcmp.Oeq
348     | A.Neq     -> L.build_fcmp L.Fcmp.One
349     | A.Less    -> L.build_fcmp L.Fcmp.Olt
350     | A.Leq     -> L.build_fcmp L.Fcmp.Ole
351     | A.Greater -> L.build_fcmp L.Fcmp.Ogt
352     | A.Geq     -> L.build_fcmp L.Fcmp.Oge
353     | A.And | A.Or ->
354         raise (Failure "internal error: semant should have rejected and/or on
    ↪ float")
355     ) e1' e2' "tmp" builder
356
357     (* String operations *)
358     | SBinop ((A.String,_ ) as e1, op, e2) ->
359       (match op with
360         A.Add     -> expr builder (A.String, SStrLit((print_sstring e1) ^ (
    ↪ print_sstring e2)))
361       | _ ->
362         let load_str e =
363           let e' = expr builder e in
364           let e_ptr = L.build_pointercast e' (L.pointer_type string_t) "strPtr"
    ↪ builder in
365           L.build_load e_ptr "strptrld" builder in
366         let s1 = load_str e1
367         and s2 = load_str e2 in
368           (match op with
369         | A.Equal   -> L.build_icmp L.Icmp.Eq
370         | A.Neq     -> L.build_icmp L.Icmp.Ne
371         | A.Less    -> L.build_icmp L.Icmp.Slt
372         | A.Leq     -> L.build_icmp L.Icmp.Sle
373         | A.Greater -> L.build_icmp L.Icmp.Sgt
374         | A.Geq     -> L.build_icmp L.Icmp.Sge
375         | _ -> raise (Failure "internal error: cannot perform this operation on
    ↪ string")
376         ) s1 s2 "strcmp" builder
377       )
378
379       (* object Null equality *)
380       | SBinop((A.Object(_),_ ) as e1, op, _) ->
381         let obj = expr builder e1 in
382         (match op with
383           A.Equal -> L.build_is_null
384         | A.Neq -> L.build_is_not_null
385         | _ -> raise (Failure "internal error: cannot perform this operation on
    ↪ object"))
386         obj "objcmp" builder
387
```

```
388       | SBinop (e1, op, e2) ->
389     let e1' = expr builder e1
390     and e2' = expr builder e2 in
391     (match op with
392       A.Add     -> L.build_add
393     | A.Sub     -> L.build_sub
394     | A.Mult    -> L.build_mul
395     | A.Div     -> L.build_sdiv
396     | A.And     -> L.build_and
397     | A.Or      -> L.build_or
398     | A.Equal   -> L.build_icmp L.Icmp.Eq
399     | A.Neq     -> L.build_icmp L.Icmp.Ne
400     | A.Less    -> L.build_icmp L.Icmp.Slt
401     | A.Leq     -> L.build_icmp L.Icmp.Sle
402     | A.Greater -> L.build_icmp L.Icmp.Sgt
403     | A.Geq     -> L.build_icmp L.Icmp.Sge
404     ) e1' e2' "tmp" builder
405       | SUnop(op, ((t, _) as e)) ->
406           let e' = expr builder e in
407     (match op with
408       A.Neg when t = A.Double -> L.build_fneg
409     | A.Neg                  -> L.build_neg
410     | A.Not                  -> L.build_not) e' "tmp" builder
411
412       | SCall ("print", [e]) ->
413         let (_, e_x) = e in
414         let e_type = find_type e_x in
415         L.build_call printf_func [| (find_str_typ e_type) ; (expr builder e) |]
416       "printf" builder
417
418       | SCall ("reverse", [e]) ->
419     L.build_call reversestring_func [| (expr builder e) |] "reverse" builder
420       | SCall ("upper", [e]) ->
421       L.build_call stringupper_func [| (expr builder e) |] "upper" builder
422       | SCall ("lower", [e]) ->
423       L.build_call stringlower_func [| (expr builder e) |] "lower" builder
424       | SCall ("substring", [e;s1;s2]) ->
425       L.build_call stringsubstring_func [| (expr builder e) ; (expr builder s1) ;
    ↪  (expr builder s2) |] "substring" builder
426       | SCall ("indexOf", [e1;e2]) ->
427       L.build_call stringindexof_func [| (expr builder e1) ; (expr builder e2) |]
    ↪  "indexOf" builder
428       | SCall ("len", [e]) ->
429       L.build_call stringlen_func [| (expr builder e) |] "len" builder
430       | SCall ("concat", [e1;e2]) ->
431       L.build_call stringconcat_func [| (expr builder e1) ; (expr builder e2) |]
    ↪ "concat" builder
432
433       | SCall ("to_string", [e]) ->
434         let e' = expr builder e in
435         let p_e = L.build_alloca (L.type_of e') "to_string" builder in
436         ignore(L.build_store e' p_e builder);
437         let v_e = L.build_bitcast p_e (string_t) "cast" builder in
438         L.build_call to_string [| (v_e) |] "to_string" builder
439
440       (* array length function *)
441       | SCall ("length", [e]) ->
```

```
442         let (ty,_) = e in
443         (match ty with
444           A.Arr(_,l) -> L.const_int i32_t l
445         | _ -> raise (Failure "function length cannot be called on non array type
    ↪ "))
446
447       | SCall (f, args) ->
448          let (fdef, fdecl) =
449           try StringMap.find f function_decls
450           with Not_found -> raise (Failure ("SCALL not found " ^ f))
451          in
452    let llargs = List.rev (List.map (expr builder) (List.rev args)) in
453    let result = (match fdecl.styp with
454                         A.Void -> ""
455                       | _ -> f ^ "_result") in
456          L.build_call fdef (Array.of_list llargs) result builder
457     in
458
459     (* LLVM insists each basic block end with exactly one "terminator"
460        instruction that transfers control.  This function runs "instr builder"
461        if the current block does not already have a terminator.  Used,
462        e.g., to handle the "fall off the end of the function" case. *)
463     let add_terminal builder instr =
464       match L.block_terminator (L.insertion_block builder) with
465   Some _ -> ()
466       | None -> ignore (instr builder) in
467
468     (* Build the code for the given statement; return the builder for
469        the statement's successor (i.e., the next instruction will be built
470        after the one generated by this call) *)
471
472     let rec stmt builder = function
473   SBlock sl -> List.fold_left stmt builder sl
474       | SExpr e -> ignore(expr builder e); builder
475       | SReturn e -> ignore(match fdecl.styp with
476                               (* Special "return nothing" instr *)
477                               A.Void -> L.build_ret_void builder
478                               (* Build return statement *)
479                             | _ -> L.build_ret (expr builder e) builder );
480                     builder
481       | SIf (predicate, then_stmt, else_stmt) ->
482          let bool_val = expr builder predicate in
483    let merge_bb = L.append_block context "merge" the_function in
484          let build_br_merge = L.build_br merge_bb in (* partial function *)
485
486    let then_bb = L.append_block context "then" the_function in
487    add_terminal (stmt (L.builder_at_end context then_bb) then_stmt)
488      build_br_merge;
489
490    let else_bb = L.append_block context "else" the_function in
491    add_terminal (stmt (L.builder_at_end context else_bb) else_stmt)
492      build_br_merge;
493
494    ignore(L.build_cond_br bool_val then_bb else_bb builder);
495    L.builder_at_end context merge_bb
496
497       | SWhile (predicate, body) ->
```

```
    let pred_bb = L.append_block context "while" the_function in
    ignore(L.build_br pred_bb builder);

    let body_bb = L.append_block context "while_body" the_function in
    add_terminal (stmt (L.builder_at_end context body_bb) body)
      (L.build_br pred_bb);

    let pred_builder = L.builder_at_end context pred_bb in
    let bool_val = expr pred_builder predicate in

    let merge_bb = L.append_block context "merge" the_function in
    ignore(L.build_cond_br bool_val body_bb merge_bb pred_builder);
    L.builder_at_end context merge_bb

      (* Implement for loops as while loops *)
      | SFor (e1, e2, e3, body) -> let e1' = SExpr e1
      in stmt builder
      ( SBlock [e1' ; SWhile (e2, SBlock [body ; SExpr e3]) ] )
    in

    (* Build the code for each statement in the function *)
    let builder = stmt builder (SBlock fdecl.sbody) in

    (* Add a return if the last block falls off the end *)
    add_terminal builder (match fdecl.styp with
        A.Void -> L.build_ret_void
      | A.Double -> L.build_ret (L.const_float double_t 0.0)
      | t -> L.build_ret (L.const_int (ltype_of_typ t) 0))
  in

  List.iter build_function_body functions;
  the_module
```

8.7 javalite.ml

```
(* Top-level of the JavaLite compiler: scan & parse the input,
   check the resulting AST and generate an SAST from it, generate LLVM IR,
   and dump the module *)

type action = Ast | Sast | LLVM_IR | Compile

let () =
  let action = ref Compile in
  let set_action a () = action := a in
  let speclist = [
    ("-a", Arg.Unit (set_action Ast), "Print the AST");
    ("-s", Arg.Unit (set_action Sast), "Print the SAST");
    ("-l", Arg.Unit (set_action LLVM_IR), "Print the generated LLVM IR");
    ("-c", Arg.Unit (set_action Compile),
      "Check and print the generated LLVM IR (default)");
  ] in
  let usage_msg = "usage: ./javalite.native [-a|-s|-l|-c] [file.jl]" in
  let channel = ref stdin in
  Arg.parse speclist (fun filename -> channel := open_in filename) usage_msg;

```

```
  let lexbuf = Lexing.from_channel !channel in
  let ast = Parser.program Scanner.token lexbuf in
  match !action with
    Ast -> print_string (Ast.string_of_program ast)
  | _ -> let sast = Semant.check ast in
    match !action with
      Ast     -> ()
    | Sast    -> print_string (Sast.string_of_sprogram sast)
    | LLVM_IR -> print_string (Llvm.string_of_llmodule (Codegen.translate sast))
    | Compile -> let m = Codegen.translate sast in
  Llvm_analysis.assert_valid_module m;
  print_string (Llvm.string_of_llmodule m)
```

8.8 stringfuncs.c

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <ctype.h>

char* reverse(const char *str)
{
    size_t len = strlen(str);
    char* revStr = (char*)malloc(len + 1);

    size_t dst, src;
    for (dst = len - 1, src = 0; src < len; src++, dst--) {
        revStr[dst] = str[src];
    }

    revStr[len] = '\0';
    return revStr;
}

char* lower(const char *str)
{
    size_t len = strlen(str);
    char* lowStr = (char*)malloc(len + 1);

    size_t src;
    for (src = len - 1, src = 0; src < len; src++) {
        lowStr[src] = tolower(str[src]);
    }

    lowStr[len] = '\0';
    return lowStr;
}

char* substring(const char *str, size_t start, size_t end)
{
    size_t len = strlen(str);
    char* sub = (char*)malloc(len + 1);

    size_t src = 0;

```

```
    while (start < end && start < len) {
        sub[src] = str[start];
        src++;
        start++;
    }

    sub[src] = '\0';
    return sub;
}

char* upper(const char *str)
{
    size_t len = strlen(str);
    char* upStr = (char*)malloc(len + 1);

    size_t src;
    for (src = len - 1, src = 0; src < len; src++) {
        upStr[src] = toupper(str[src]);
    }

    upStr[len] = '\0';
    return upStr;
}

int indexOf(const char *str, const char *find)
{
    int i;
    size_t len = strlen(str);

    for (i = 0; i < len; i++)
    {
        if (str[i] == *find)
            return i;
    }

    return -1;
}

int len(const char *str)
{
    int l;
    size_t len = strlen(str);

    // l = (int*)(len);
    l = (int)(len);
}

char* concat(const char *str1, const char *str2)
{
    size_t len1 = strlen(str1);
    size_t len2 = strlen(str2);
    char* both = (char*)malloc(len1 + len2 + 1);

    strcpy(both, str1);
    strcat(both, str2);

    return both;
```

```
}
```

8.9 Makefile

```
# "make test" Compiles everything and runs the regression tests

.PHONY : test
test : all testall.sh
  ./testall.sh

# "make all" builds the executable as well as the built-in libraries designed
# to test linking external code

.PHONY : all
all : javalite.native stringfuncs.o

# "make javalite.native" compiles the compiler
#
# The _tags file controls the operation of ocamlbuild, e.g., by including
# packages, enabling warnings
#
# See https://github.com/ocaml/ocamlbuild/blob/master/manual/manual.adoc

javalite.native :
  opam config exec -- \
  ocamlbuild -use-ocamlfind javalite.native

# "make clean" removes all generated files

.PHONY : clean
clean :
  ocamlbuild -clean
  rm -rf testall.log BuildAndRun.log ocamlllvm *.diff *.exe *.ll *.s *.out *.err
     ↪ *.o


# Testing the "string" example
stringfuncs : stringfuncs.c
  cc -o string -DBUILD_TEST stringfuncs.c

# Building the tarball

TESTS = \
  add1 arith1 arith2 arith3 fib float1 float2 float3 for1 for2 func1 \
  func2 func3 func4 func5 func6 func7 func8 func9 gcd2 gcd global1 \
  global2 global3 hello if1 if2 if3 if4 if5 if6 local1 local2 ops1 \
  ops2 var1 var2 while1 while2

FAILS = \
  assign1 assign2 assign3 dead1 dead2 expr1 expr2 expr3 float1 float2 \
  for1 for2 for3 for4 for5 func1 func2 func3 func4 func5 func6 func7 \
  func8 func9 global1 global2 if1 if2 if3 nomain printb print \
  return1 return2 while1 while2

TESTFILES = $(TESTS:%=test-%.jl) $(TESTS:%=test-%.out) \
```

```
      $(FAILS:%=fail-%.jl) $(FAILS:%=fail-%.err)

TARFILES = ast.ml sast.ml codegen.ml Makefile _tags javalite.ml parser.mly \
  README scanner.mll semant.ml testall.sh \
  arcade-font.pbm font2c \
  Dockerfile \
  $(TESTFILES:%=tests/%)

javalite.tar.gz : $(TARFILES)
  cd .. && tar czf javalite/javalite.tar.gz \
    $(TARFILES:%=javalite/%)
```

8.10 testall.sh

```
#!/bin/sh

# Regression testing script for JavaLite
# Step through a list of files
#  Compile, run, and check the output of each expected-to-work test
#  Compile and check the error of each expected-to-fail test

# Path to the LLVM interpreter
LLI="lli"
#LLI="/usr/local/opt/llvm/bin/lli"

# Path to the LLVM compiler
LLC="llc"

# Path to the C compiler
CC="cc"

# Path to the javalite compiler.  Usually "./javalite.native"
# Try "_build/javalite.native" if ocamlbuild was unable to create a symbolic link
    ↪ .
JAVALITE="./javalite.native"
#JAVALITE="_build/javalite.native"

# Set time limit for all operations
ulimit -t 30

globallog=testall.log
rm -f $globallog
error=0
globalerror=0

keep=0

Usage() {
    echo "Usage: testall.sh [options] [.jl files]"
    echo "-k    Keep intermediate files"
    echo "-h    Print this help"
    exit 1
}

SignalError() {
```

```
    if [ $error -eq 0 ] ; then
  echo "FAILED"
  error=1
    fi
    echo "  $1"
}

# Compare <outfile> <reffile> <difffile>
# Compares the outfile with reffile.  Differences, if any, written to difffile
Compare() {
    generatedfiles="$generatedfiles $3"
    echo diff -b $1 $2 ">" $3 1>&2
    diff -b "$1" "$2" > "$3" 2>&1 || {
  SignalError "$1 differs"
  echo "FAILED $1 differs from $2" 1>&2
    }
}

# Run <args>
# Report the command, run it, and report any errors
Run() {
    echo $* 1>&2
    eval $* || {
  SignalError "$1 failed on $*"
  return 1
    }
}

# RunFail <args>
# Report the command, run it, and expect an error
RunFail() {
    echo $* 1>&2
    eval $* && {
  SignalError "failed: $* did not report an error"
  return 1
    }
    return 0
}

Check() {
    error=0
    basename=`echo $1 | sed 's/.*\\///
                              s/.jl//'`
    reffile=`echo $1 | sed 's/.jl$//'`
    basedir="`echo $1 | sed 's/\/[^\/]*$//'`/."

    echo -n "$basename..."

    echo 1>&2
    echo "###### Testing $basename" 1>&2

    generatedfiles=""

    generatedfiles="$generatedfiles ${basename}.ll ${basename}.s ${basename}.exe
    ↪ ${basename}.out" &&
    Run "$JAVALITE" "$1" ">" "${basename}.ll" &&
    Run "$LLC" "-relocation-model=pic" "${basename}.ll" ">" "${basename}.s" &&
```

```
    Run "$CC" "-o" "${basename}.exe" "${basename}.s" "stringfuncs.o" &&
    Run "./${basename}.exe" > "${basename}.out" &&
    Compare ${basename}.out ${reffile}.out ${basename}.diff

    # Report the status and clean up the generated files

    if [ $error -eq 0 ] ; then
  if [ $keep -eq 0 ] ; then
      rm -f $generatedfiles
  fi
  echo "OK"
  echo "###### SUCCESS" 1>&2
    else
  echo "###### FAILED" 1>&2
  globalerror=$error
    fi
}

CheckFail() {
    error=0
    basename=`echo $1 | sed 's/.*\\///
                             s/.jl//'`
    reffile=`echo $1 | sed 's/.jl$//'`
    basedir="`echo $1 | sed 's/\/[^\/]*$//'`/."

    echo -n "$basename..."

    echo 1>&2
    echo "###### Testing $basename" 1>&2

    generatedfiles=""

    generatedfiles="$generatedfiles ${basename}.err ${basename}.diff" &&
    RunFail "$JAVALITE" "<" $1 "2>" "${basename}.err" ">>" $globallog &&
    Compare ${basename}.err ${reffile}.err ${basename}.diff

    # Report the status and clean up the generated files

    if [ $error -eq 0 ] ; then
  if [ $keep -eq 0 ] ; then
      rm -f $generatedfiles
  fi
  echo "OK"
  echo "###### SUCCESS" 1>&2
    else
  echo "###### FAILED" 1>&2
  globalerror=$error
    fi
}

while getopts kdpsh c; do
    case $c in
  k) # Keep intermediate files
      keep=1
      ;;
  h) # Help
      Usage
```

```
      ;;
    esac
done

shift `expr $OPTIND - 1`

LLIFail() {
  echo "Could not find the LLVM interpreter \"$LLI\"."
  echo "Check your LLVM installation and/or modify the LLI variable in testall.sh "
  exit 1
}

which "$LLI" >> $globallog || LLIFail


if [ ! -f stringfuncs.o ]
then
    echo "Could not find stringfuncs.o"
    echo "Try \"make stringfuncs.o\""
    exit 1
fi

# if [ ! -f arrayfuncs.o ]
# then
#     echo "Could not find arrayfuncs.o"
#     echo "Try \"make arrayfuncs.o\""
#     exit 1
# fi


if [ $# -ge 1 ]
then
    files=$@
else
    files="tests/test-*.jl tests/fail-*.jl"
fi

for file in $files
do
    case $file in
  *test-*)
      Check $file 2>> $globallog
      ;;
  *fail-*)
      CheckFail $file 2>> $globallog
      ;;
  *)
      echo "unknown file type $file"
      globalerror=1
      ;;
    esac
done

exit $globalerror
```

8.11 Testing Files

8.11.1 Test Cases

test-add1.jl

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


print( add(17, 25) );
```

test-arith1.jl

```
print(39 + 3);
```

test-arith2.jl

```
print(1 + 2 * 3 + 4);
```

test-arith3.jl

```
int foo(int a)
{
  return a;
}

int a = 42;
a = a + 5;
print(a);
```

test-arith4.jl

```
print(3/2);
```

test-array1.jl

```
int[] intArr = [1,2,3];
int i = intArr[2];
print(i);
string[] strArr = ["h", "e", "y"];
for (i = 0 ; i < length(strArr) ; i = i + 1) {
   print(strArr[i]);
}
```

test-array2.jl

```
int[] intArr = [1, 3, 5, 7];
int i = 0;
int j = 0;
while (i < length(intArr)) {
  print(j);
  j = j + intArr[i];
  i = i + 1;
}
print(j);
```

test-array3.jl

```
string[] strArr1 = ["te", "st"];
string[] strArr2 = ["ar", "ry", "3"];
print(strArr2[1]);
string[][] str2dArr = [strArr1, strArr2];
strArr2 = str2dArr[0];
print(strArr2[1]);
int[][] int2dArr = [[1,2,3],[1,2]];
int[] intArr = int2dArr[1];
print(intArr[0]);
```

test-array4.jl

```
void addOne(int[] arr, int len) {
    for (int i = 0; i < len; i = i + 1) {
        arr[i] = arr[i] + 1;
    }
}


int[] intArr = [1,2,3];
print(intArr[0]);
print(intArr[1]);
print(intArr[2]);
addOne(intArr, 3);
print(intArr[0]);
print(intArr[1]);
print(intArr[2]);
```

test-array5.jl

```
void repeatStr(string[] arr, int len) {
    for (int i = 0; i < len; i = i + 1) {
        arr[i] = concat(arr[i], arr[i]);
    }
}


string[] strArr = ["hi", "hello", "hey"];
print(strArr[0]);
print(strArr[1]);
print(strArr[2]);
repeatStr(strArr, 3);
print(strArr[0]);
print(strArr[1]);
print(strArr[2]);
```

test-array6.jl

```
int[] createArr() {
    int[] nums = [1,2,3];
    return nums;
}
int[] ns = [3,4];
print(ns[1]);
ns = createArr();
print(ns[1]);
```

test-array-length1.jl

```
int[] nums = [1,2,42,4,43];
int i = length(nums);
print(i);
```

test-array-length2.jl

```
string[] strArr1 = ["h", "e", "y"];
int l1 = length(strArr1);
print(l1);
string[] strArr2 = ["hello", "world"];
int l2 = length(strArr2);
print(l2);
string[][] str2dArr = [strArr1, strArr2];
int l2d = length(str2dArr);
print(l2d);
```

test-bubble-sort.jl

```
int[] sortingArr = [52,14,72,5,66,7,12,31,9,3,54,41,53,12,61];
int i = 0;
int j = 0;
int tmpForSwap = 0;

int length = length(sortingArr);

for (i = 0 ; i < length-1 ; i = i + 1) {
for (j =0; j< length-i-1;j = j + 1){
  if (sortingArr[j] > sortingArr[j+1]){

    tmpForSwap = sortingArr[j];
    sortingArr[j] = sortingArr[j+1];
    sortingArr[j+1] = tmpForSwap;
  }
}
}

for (i = 0 ; i < length; i = i + 1) {
   print(sortingArr[i]);
}
```

test-class1.jl

```
class square {
    string name;
    int side;
}


class square sq = square("fst_sq", 42);
print(sq.side);
print(sq.name);
sq.side = 2;
sq.name = "snd_sq";
print(sq.side);
print(sq.name);
```

test-class2.jl

```
class Person {
    string name;
    int age;
    string phrase;
}

void sayhi(class Person p) {
    string n = concat(p.name, " say:");
    print(n);
    string s = p.phrase;
    for (int i = 0; i < p.age; i = i + 1) {
        s = concat(s, p.phrase);
    }
    print(s);
}


class Person alice = Person("Alice", 3, "hey");
sayhi(alice);
```

test-class3.jl

```
class Item {
    string data;
    int priority;
}

class Item getHighestPriority(class Item[] arr, int len) {
    class Item hi = arr[0];
    for (int i = 1; i < len; i = i + 1) {
        class Item ci = arr[i];
        if (ci.priority > hi.priority) {
            hi = ci;
        }
    }
    return hi;
}

// may want to add "insert(data, priority)" , "deleteHighestPriority()"
// after implemented array functions



class Item i1 = Item("medium", 1);
class Item i2 = Item("high", 2);
class Item i3 = Item("extra high", 3);
class Item i4 = Item("low", 0);
class Item[] itemArr = [i1, i2, i3, i4];
class Item hi = getHighestPriority(itemArr, 4);
print(hi.priority);
print(hi.data);
```

test-class4.jl

```
class Record {
    int index;
```

```
    int data;
    string comment;
}

void updateComment(class Record r, string cmt) {
    string msg = concat("old comment: ", r.comment);
    print(msg);
    r.comment = cmt;
}

class Record rec = Record(0, 42, "added on Apr 10, 2021");
updateComment(rec, "updated on Apr 18, 2021");
string msg = concat("new comment: ", rec.comment);
print(msg);
```

test-class5.jl

```
class Course {
    string name;
    string department;
}

class Person {
    string name;
    int id;
    class Course[] take_courses;
}

void printCourses(class Person p, int len) {
    class Course[] courses = p.take_courses;
    for (int i = 0; i < len; i = i + 1) {
        class Course curr_course = courses[i];
        string fullname = concat(curr_course.department, curr_course.name);
        print(fullname);
    }
}

void updateDepartment(class Person p, int len) {
    class Course[] courses = p.take_courses;
    for (int i = 0; i < len; i = i + 1) {
        class Course curr_course = courses[i];
        if (curr_course.department != "COMS") {
            curr_course.department = "NO";
        }
    }
}

class Course cs4115 = Course("PLT", "COMS");
class Course m4042 = Course("Algebra", "MATH");
class Course cs6111 = Course("Database", "COMS");
class Person jess = Person("Jesse", 3222, [cs4115, m4042, cs6111]);
print(jess.id);
printCourses(jess, 3);
updateDepartment(jess, 3);
printCourses(jess, 3);
```

test-class6.jl

```
class Node {
    int data;
    class Node next;
}

void push(class Node head, int data) {
    class Node n = head;
    while (n.next != null) {
        n = n.next;
    }
    n.next = Node(data, null);
}

int pop(class Node head) {
    class Node n = head.next;
    int data = head.data;
    if (n == null) {
        head = null;
    } else {
        class Node prev = head;
        while (n.next != null) {
            n = n.next;
            prev = prev.next;
        }
        data = n.data;
        prev.next = null;
    }
    return data;
}

void printNodes(class Node head) {
    class Node n = head;
    while (n.next != null) {
        print(n.data);
        n = n.next;
    }
    print(n.data);
}

class Node head = Node(42, null);
printNodes(head);
int[] dataArr = [3,4,5];
push(head, dataArr[0]);
push(head, dataArr[1]);
push(head, dataArr[2]);
printNodes(head);
pop(head);
printNodes(head);
```

test-counting-sort.jl

```
int[] countingSortArray = [0,0,0,0,0,0,0,0,0,0
            ,0,0,0,0,0,0,0,0,0,0
            ,0,0,0,0,0,0,0,0,0,0
            ,0,0,0,0,0,0,0,0,0,0
            ,0,0,0,0,0,0,0,0,0,0
            ];
```

```
int countingSortArrayLength = length(countingSortArray);
int[] sortingArr = [32,17,31,5,26,13,47,31,9,3,24,41,43,12,11];
int length = length(sortingArr);
int i = 0;
int j = 0;

for (i = 0 ; i < length ; i = i + 1) {
  countingSortArray[sortingArr[i]] = countingSortArray[sortingArr[i]] + 1;
}

for (i = 0 ; i < countingSortArrayLength ; i = i + 1) {
   if(countingSortArray[i] > 0){
    for(j=0;j<countingSortArray[i];j = j+1){
      print(i);
    }
   }
}
```

test-double1.jl

```
double a = 3.14159267;
print(a);
```

test-double2.jl

```
double a = 3.14159267;
double b = -2.71828;
double c = a + b;
print(c);
```

test-double3.jl

```
void testdouble(double a, double b)
{
  print(a + b);
  print(a - b);
  print(a * b);
  print(a / b);
  print(a == b);
  print(a == a);
  print(a != b);
  print(a != a);
  print(a > b);
  print(a >= b);
  print(a < b);
  print(a <= b);
}

double c = 42.0;
double d = 3.14159;

testdouble(c, d);

testdouble(d, d);
```

test-fib.jl

```
int fib(int x)
```

```
{
  if (x < 2) return 1;
  return fib(x-1) + fib(x-2);
}

print(fib(0));
print(fib(1));
print(fib(2));
print(fib(3));
print(fib(4));
print(fib(5));
```

test-for1.jl

```
for (int i = 0 ; i < 5 ; i = i + 1) {
  print(i);
}
print(42);
```

test-for2.jl

```
int i = 0;
for ( ; i < 5; ) {
  print(i);
  i = i + 1;
}
print(42);
```

test-func-string1.jl

```
string s = "Hello World";
string r = reverse(s);
print(s);
print(r);
r = reverse(r);
print(r);

s = "Foo Foo Doo Bar";
string u = upper(s);
print(s);
print(u);
```

test-func-string2.jl

```
string x = "ThIS is A vERy loNG SENTence THat I AM writing";
string l = lower(x);
int len_x = len(x);

print(x);
print(l);
print(len_x);
```

test-func-string3.jl

```

string s = "Hello World!!";
string t = "FooFooDooDooBar";

```

```
print(s);
string r = substring(s,1,5);
print(r);

r = substring(s,4,7);
print(r);

r = substring(t,0,10);
print(r);

r = substring(t,4,6);
print(r);
```

test-func-string4.jl

```
string s = "Hello World!!";
string t = "FooFooDooDooBar";

int si = indexOf(s,"e");
print(si);

int ti = indexOf(t,"D");
print(ti);

int sn = indexOf(s,"D");
print(sn);

string c = concat(s,t);
print(c);

s = substring(s,0,6);
c = concat(s,t);
c = concat(c,"!!!!!");
print(c);
```

test-func1.jl

```
int add(int a, int b)
{
  return a + b;
}


int a = add(39, 3);
print(a);
```

test-func2.jl

```
/* Bug noticed by Pin-Chin Huang */

int fun(int x, int y)
{
  return 0;
}

int i = 1;

fun(i = 2, i = i+1);
```

```
11
12 print(i);
```

test-func3.jl

```
1  void printem(int a, int b, int c, int d)
2  {
3    print(a);
4    print(b);
5    print(c);
6    print(d);
7  }
8
9
10 printem(42,17,192,8);
```

test-func4.jl

```
1  int add(int a, int b)
2  {
3    int c = a + b;
4    return c;
5  }
6
7
8  int d = add(52, 10);
9  print(d);
```

test-func5.jl

```
1  int foo(int a)
2  {
3    return a;
4  }
```

test-func6.jl

```
1  void foo() {}
2
3  int bar(int a, bool b, int c) { return a + c; }
4
5
6  print(bar(17, false, 25));
```

test-func7.jl

```
1  void foo(int a)
2  {
3    print(a + 3);
4  }
5
6
7  foo(40);
```

test-func8.jl

```
1  void foo(int a)
2  {
```

```
  print(a + 3);
  return;
}

foo(40);
```

test-functional1.jl

```
int x = 2;
print(x);
```

test-functional2.jl

```
int x = 0;
print(x);

int update(int x)
{
    x = x + 1;
    return x;
}

x = update(x);
print(x);
```

test-functional3.jl

```
void print1()
{
    print("hello");
}

print(1);

void print2()
{
    print("hi");
}

print(2);

void doNothing() {}

doNothing();
print1();
print2();
```

test-gcd1.jl

```
int gcd(int a, int b) {
  while (a != b) {
    if (a > b) a = a - b;
    else b = b - a;
  }
  return a;
}

print(gcd(2,14));
```

```
print(gcd(3,15));
print(gcd(99,121));
```

test-gcd2.jl

```
int gcd(int a, int b) {
  while (a != b)
    if (a > b) a = a - b;
    else b = b - a;
  return a;
}

print(gcd(14,21));
print(gcd(8,36));
print(gcd(99,121));
```

test-if1.jl

```
if (true) print(42);
print(17);
```

test-if2.jl

```
if (true) print(42); else print(8);
print(17);
```

test-if3.jl

```
if (false) print(42);
print(17);
```

test-if4.jl

```
if (false) print(42); else print(8);
print(17);
```

test-if5.jl

```
int cond(bool b)
{
  int x = 0;
  if (b)
    x = 42;
  else
    x = 17;
  return x;
}


print(cond(true));
print(cond(false));
```

test-if6.jl

```
int cond(bool b)
{
  int x = 10;
  if (b)
```

```
    if (x == 10)
      x = 42;
  else
    x = 17;
  return x;
}


print(cond(true));
print(cond(false));
```

test-local1.jl

```
void foo(bool i)
{
  int i = 42; /* Should hide the formal i */
  print(i + i);
}


foo(true);
```

test-local2.jl

```
int foo(int a, bool b)
{
  int c = a;
  bool d = false;
  return c + 10;
}

print(foo(37, false));
```

test-ops1.jl

```
print(1 + 2);
print(1 - 2);
print(1 * 2);
print(100 / 2);
print(99);
print(1 == 2);
print(1 == 1);
print(99);
print(1 != 2);
print(1 != 1);
print(99);
print(1 < 2);
print(2 < 1);
print(99);
print(1 <= 2);
print(1 <= 1);
print(2 <= 1);
print(99);
print(1 > 2);
print(2 > 1);
print(99);
print(1 >= 2);
print(1 >= 1);
```

```
print(2 >= 1);
```

test-ops2.jl

```
print(true);
print(false);
print(true && true);
print(true && false);
print(false && true);
print(false && false);
print(true || true);
print(true || false);
print(false || true);
print(false || false);
print(!false);
print(!true);
print(-10);
print(--42);
```

test-print.jl

```
int i = 3;
int j = 9;
print(i + j);
print(i - j);
print(i / j);

string s = "Foo";
print(s);

double a = 3.14159267;
double b = -2.71828;
print(a + b);
print(a == b);

print(true && false);
print(!false);
print("hello" + " world");
print(1 + 2 * 3 + 4);

string[] strArr = ["hello", "world"];
print(strArr[0]);

int fib(int x)
{
  if (x < 2) return 1;
  return fib(x-1) + fib(x-2);
}
print(fib(0));
print(fib(3));
```

test-string-binop1.jl

```
string c = "Hello " + "FooFooDooDooBar";
print(c);

c = "Foo " + "Bar";
print(c);
```

test-string-binop2.jl

```
string s1 = "hello";
string s2 = "hey";
string s3 = "hello";
print(s1 == s2); // 0
print(s1 == s3); // 1
print(s1 != s2); // 1
print(s1 > s2); // 1
print(s1 < s2); // 0
print(s1 >= s2); // 1
print(s1 <= s3); // 1
```

test-var1.jl

```
int a = 42;
print(a);
```

test-var2.jl

```
string s = "hello";
int a = 1;
print(s);
print(a);
```

test-while1.jl

```
int i = 5;
while (i > 0) {
  print(i);
  i = i - 1;
}
print(42);
```

test-while2.jl

```
int foo(int a)
{
  int j = 0;
  while (a > 0) {
    j = j + 2;
    a = a - 1;
  }
  return j;
}


print(foo(7));
```

8.11.2 Fail Cases

fail-array1.jl

```
int[] nums = ["h", "e", "l", "l", "o"];
```

fail-array2.jl

```
string[] arr = ["h", "e", 1, "l", "o"];
```

fail-array3.jl

```
int nums = 1;
string[] arr = ["h", "e", "l", "l", "o"];
string se = arr[2];
int ie = nums[1];
```

fail-array4.jl

```
string[] strArr2 = ["l", "l", "o"];
strArr2[2] = 1;
```

fail-array5.jl

```
int[] int2dArr = [[1,2,3],[4,5]];
```

fail-array-length.jl

```
string s = "hello world";
int i = length(s);
print(i);
```

fail-assign1.jl

```
int i = 42;
bool b = true;
b = false;
i = false; /* Fail: assigning a bool to an integer */
```

fail-assign2.jl

```
int i = 0;
bool b = 48; /* Fail: assigning an integer to a bool */
```

fail-assign3.jl

```
void myvoid()
{
  return;
}


int i = myvoid(); /* Fail: assigning a void to an integer */
```

fail-class1.jl

```
class square {
    string name;
    int side;
}

class square sq = square("fst_sq", "42");
int s = sq.side;
string n = sq.name;
```

fail-class2.jl

```
class square {
    string name;
    int side;
}

class square sq = square("fst_sq", 42);
int s = sq.side;
int n = sq.name;
```

fail-dead1.jl

```
int i = 15;
return i;
i = 32; /* Error: code after a return */
```

fail-dead2.jl

```
int i = 0;

{
  i = 15;
  return i;
}
i = 32; /* Error: code after a return */
```

fail-double1.jl

```
-3.5 && 1; /* Double with AND? */
```

fail-double2.jl

```
-3.5 && 2.5; /* Double with AND? */
```

fail-expr1.jl

```
void foo(int c, bool d)
{
  int dd = 0;
  bool e = true;
  e + dd; /* Error: bool + int */
}
```

fail-expr2.jl

```
void foo(int c, bool d)
{
  int d = 0;
  bool e = false;
  e + d; /* Error: bool + int */
}
```

fail-expr3.jl

```
void foo(int c, double d)
{
  int d = 0;
```

```
  double e = 0.0;
  e + d; /* Error: double + int */
}
```

fail-for1.jl

```
int i = 0;
int k = 0;
for ( ; true ; ) {} /* OK: Forever */

for (i = 0 ; i < 10 ; i = i + 1) {
  if (i == 3) {
    j = j + 1;
  }
}

for (j = 0; i < 10 ; i = i + 1) {} /* j undefined */
```

fail-for2.jl

```
int i = 0;

for (i = 0; j < 10 ; i = i + 1) {} /* j undefined */
```

fail-for3.jl

```
int i = 0;
for (i = 0; i ; i = i + 1) {} /* i is an integer, not Boolean */
```

fail-for4.jl

```
int i = 0;

for (i = 0; i < 10 ; i = j + 1) {} /* j undefined */
```

fail-for5.jl

```
int i = 0;

for (i = 0; i < 10 ; i = i + 1) {
  foo(); /* Error: no function foo */
}
```

fail-func1.jl

```
int foo() {}

int bar() {}

int baz() {}

void bar() {} /* Error: duplicate function bar */
```

fail-func2.jl

```
int foo(int a, bool b, int c) { }

void bar(int a, bool b, int a) {} /* Error: duplicate formal a in bar */
```

fail-func3.jl

```
int foo(int a, bool b, int c) { }

void bar(int a, void b, int c) {} /* Error: illegal void formal b */
```

fail-func4.jl

```
int foo() {}

void bar() {}

int print() {} /* Should not be able to define print */

void baz() {}
```

fail-func5.jl

```
void foo(int a, bool b)
{
}


foo(42, true);
foo(42); /* Wrong number of arguments */
```

fail-func6.jl

```
void foo(int a, bool b)
{
}


foo(42, true);
foo(42, 42); /* Fail: int, not bool */
```

fail-func7.jl

```
void foo(int a, bool b)
{
}


foo(42, true);
foo(42, true, false); /* Wrong number of arguments */
```

fail-func8.jl

```
void foo(int a, bool b)
{
}

void bar()
{
}


foo(42, true);
foo(42, bar()); /* int and void, not int and bool */
```

fail-functional1.jl

```
int x = 0;

void update()
{
    x = x + 1;
}

update();
print(x);
```

fail-if1.jl

```

if (true) {}
if (false) {} else {}
if (42) {} /* Error: non-bool predicate */
```

fail-if2.jl

```
if (true) {
  foo; /* Error: undeclared variable */
}
```

fail-if3.jl

```
if (true) {
  42;
} else {
  bar; /* Error: undeclared variable */
}
```

fail-print1.jl

```
/* Should be illegal to redefine */
void print() {}
```

fail-print2.jl

```
string[] strArr = ["hello", "world"];
print(strArr);
```

fail-return1.jl

```
int foo(){
  return 12;
  int a=2;
}
```

fail-return2.jl

```
void foo()
{
  if (true) return 42; /* Should return void */
  else return;
}
```

fail-return3.jl

```
int x = 0;
return x;
```

fail-string-binop1.jl

```
string s1 = "hello";
string s2 = "hey";
string s3 = "hello";
print(s1 - s2);
```

fail-while1.jl

```
int i = 0;

while (true) {
  i = i + 1;
}

while (42) { /* Should be boolean */
  i = i + 1;
}
```

fail-while2.jl

```
int i = 0;

while (true) {
i = i + 1;
}

while (true) {
  foo(); /* foo undefined */
}
```

8.11.3 Performance Tests

test-Array-Access.jl

```
int[] Arr = [52,14,72,5,66,7,12,31,9,3,54,41,53,12,61];
int length = 15;

int count = 100000000;
int i=0;

int v = 0;
while(count > 0){
      for(i=0;i<length;i = i+1){
        v = Arr[i];
        Arr[i] = 3;
      }
      count = count-1;
}

print("Array-Access Test Completed");
```

test-Class-Field-Access.jl

```
class Person {
    string name;
    int age;
    string phrase;
}
int count = 200000000;

class Person alice = Person("Alice", 3, "hey");
while(count > 0){
      string s = alice.phrase;
      s = alice.phrase;
      count = count-1;
}

print("Class-Field-Access Test Completed");
```

test-Double-Arithmetic.jl

```
double AddTest = 0.0;
double MinusTest = 2100000005.0;
double MultiplyTest = 2.0;
double DivideTest = 2.0;

int count = 2100000005;

while(count > 0){

      AddTest = AddTest + 1.0;
      MinusTest = MinusTest - 1.0;
      MultiplyTest = 3.0 * 3.0;
      DivideTest = 6.0/2.0;

      count = count-1;
}

print("Double-Arithmetic Test Completed");
```

test-If-Condition.jl

```
int count = 2100000005;
int n=0;

while(count > 0){
      if(1 < 3){
        n = 2;
      }

      count = count-1;
}

print("If-Condition Test Completed");
```

test-Int-Arithmetic.jl

```
int AddTest = 0;
int MinusTest = 2100000005;
```

```
int MultiplyTest = 2;
int DivideTest = 2;

int count = 2100000005;

while(count > 0){
      AddTest = AddTest + 1;
      MinusTest = MinusTest - 1;
      MultiplyTest = 3*3;
      DivideTest = 6/2;

      count = count-1;
}

print("Int-Arithmetic Test Completed");
```

8.11.4 Demo Files

demo-arrays.jl

```
int[][] a = [[2, 5], [1, 2]];
int[][] b = [[4, 5], [2, 3]];
deterMult(a, b);

// assuming size of 2s
void deterMult(int[][] a, int[][] b) {
    int[] row1A = a[0];
    int[] row2A = a[1];
    int[] row1B = b[0];
    int[] row2B = b[1];

    int f0 = (row1A[0] * row1B[0]) + (row1A[1] * row2B[0]);
    int f1 = (row1A[0] * row1B[1]) + (row1A[1] * row2B[1]);
    int f2 = (row2A[0] * row1B[0]) + (row2A[1] * row2B[0]);
    int f3 = (row2A[0] * row1B[1]) + (row2A[1] * row2B[1]);
    print((f0 * f3) - (f1 * f2));
}

/*
|[ 2 5 ][ 4 5 ]|   |[ 18 25 ]|
|[ 1 2 ][ 2 3 ]| = |[ 8  11 ]| = (198 - 200) = -2 */
```

demo-class.jl

```
class Course {
    string name;
    string department;
}

class Student {
    string name;
    int id;
    string class_of;
    class Course[] courses_taken;
}

```

```
void printCourses(class Student s, int len) {
    class Course[] courses = s.courses_taken;
    for (int i = 0; i < len; i = i + 1) {
        class Course curr_course = courses[i];
        string fullname = concat(curr_course.department, curr_course.name);
        print(fullname);
    }
}

void updateDepartment(class Student s, int len) {
    class Course[] courses = s.courses_taken;
    for (int i = 0; i < len; i = i + 1) {
        class Course curr_course = courses[i];
        if (curr_course.department != "COMS") {
            curr_course.department = "NOTCS! --- ";
        }
    }
}

void prettyPrint(class Student s) {
    string j = upper(s.name);
    print(j);
    print(s.id);
    string c = concat("Class Of: ", s.class_of);
    print(c);
    print("----------");
    printCourses(s, 3);

    print("----------");
    updateDepartment(s, 3);
    printCourses(s, 3);
}

class Course coms4115 = Course("PLT", "COMS");
class Course math4042 = Course("Algebra", "MATH");
class Course coms6111 = Course("Database", "COMS");

class Student jess = Student("Jesse Blah", 5555, "Spring 2022", [coms4115,
    ↪ math4042, coms6111]);

prettyPrint(jess);
```

demo-linkedlist.jl

```
class Node {
    int data;
    class Node next;
}

void push(class Node head, int data) {
    class Node n = head;
    while (n.next != null) {
        n = n.next;
    }
    n.next = Node(data, null);
}

int pop(class Node head) {
```

```
    class Node n = head.next;
    int data = head.data;
    if (n == null) {
        head = null;
    } else {
        class Node prev = head;
        while (n.next != null) {
            n = n.next;
            prev = prev.next;
        }
        data = n.data;
        prev.next = null;
    }
    return data;
}


// intialize head
class Node head = Node(42, null);
printNodes(head);

// push new nodes
int[] dataArr = [3,4,5];
for (int i = 0; i < length(dataArr); i = i + 1) {
    push(head, dataArr[i]);
}

printNodes(head);

// pop the last node
print("popped:");
print(pop(head));
print("");
printNodes(head);


void printNodes(class Node head) {
    print("list:");
    class Node n = head;
    while (n.next != null) {
        print(n.data);
        n = n.next;
    }
    print(n.data);
    print("");
}
```

9 Project Log

```
commit 354e3a465987be2a29d3a44d8b91dc5cb2056e0f
Author: HongfeiChenCU <hc3222@columbia.edu>
Date:   Mon Apr 26 14:31:09 2021 -0700

    memory allocation updated
```

```
 codegen.ml                   | 11 +++++++++---
 dev_tests/arrayinc.c         |  7 ++++++-
 dev_tests/ast-test-arrays.jl | 45 ++++++++++++++++++++++++++++++++++++++++++---
 semant.ml                    | 21 +++++++++++++++++----
 tests/test-array6.jl         |  8 ++++++++
 tests/test-array6.out        |  2 ++
 6 files changed, 75 insertions(+), 19 deletions(-)

commit 3fd63c280cc3a6857f2296a12992031695d42227
Merge: d0c80fa f01752e
Author: HongfeiChenCU <hc3222@columbia.edu>
Date:   Sun Apr 25 20:50:20 2021 -0700

    Merge branch 'main' of https://github.com/FranCao/javalite into main

commit d0c80fa9336c6b12c21dde7af9717c36c0aa7b97
Author: HongfeiChenCU <hc3222@columbia.edu>
Date:   Sun Apr 25 20:50:10 2021 -0700

    linkedlist demo

 demo/demo-linkedlist.jl  | 61 ++++++++++++++++++++++++++++++++++++++++++++++++++++++++
 demo/demo-linkedlist.out |  8 +++++++
 2 files changed, 69 insertions(+)

commit f01752e4d3221b36b97d124f03122fa42ee96833
Author: Mateo Maturana <mateomaturana@yahoo.com>
Date:   Sun Apr 25 23:48:11 2021 -0400

    demo arrays

 demo/demo-arrays.jl | 11 ++----------
 1 file changed, 2 insertions(+), 9 deletions(-)

commit 9936704c941b728a0549ebf530ea8a05b7678427
Author: Mateo Maturana <mateomaturana@yahoo.com>
Date:   Sun Apr 25 23:46:40 2021 -0400

    added 2d array demo code

 demo/demo-arrays.jl | 28 ++++++++++++++++++++++++++++
 1 file changed, 28 insertions(+)

commit 1cddf7f8af2e080cfe8c2467c41a94ed5929f561
Author: Frances Cao <frances.te.cao@gmail.com>
Date:   Sun Apr 25 21:29:16 2021 -0400

    finalize class demo

 demo/demo-class.jl  | 16 ++++++++++------
 demo/demo-class.out |  9 ---------
 2 files changed, 10 insertions(+), 15 deletions(-)

commit d7bf501e49d0f14555937660fe362c0815edb4ed
Author: HongfeiChenCU <hc3222@columbia.edu>
Date:   Sun Apr 25 15:23:49 2021 -0700
```

```
63
64      codegen cleanup
65
66   codegen.ml | 47 ----------------------------------------------
67   1 file changed, 47 deletions(-)
68
69  commit 3117364856185177531cd3722cb181f35ecdcab6
70  Author: HongfeiChenCU <hc3222@columbia.edu>
71  Date:   Sun Apr 25 15:22:18 2021 -0700
72
73      array length, 2d array fix
74
75   Makefile                     |  6 +-----
76   ast.ml                       |  2 +-
77   codegen.ml                   | 45 ++++++++++++++++++++++----------------
78   dev_tests/ast-test-arrays.jl |  8 ++++----
79   semant.ml                    | 15 ++++++++++---
80   tests/fail-array1.err        |  2 +-
81   tests/fail-array5.err        |  1 +
82   tests/fail-array5.jl         |  1 +
83   tests/test-array3.jl         |  3 +++
84   tests/test-array3.out        |  3 ++-
85   10 files changed, 49 insertions(+), 37 deletions(-)
86
87  commit 4f7914b39a0ed564b07164955ce275013adc3e89
88  Author: Frances Cao <frances.te.cao@gmail.com>
89  Date:   Sun Apr 25 15:42:34 2021 -0400
90
91      trying printArr
92
93   codegen.ml                   | 34 ++++++++++++++++++++++++++++++++--
94   dev_tests/ast-test-arrays.jl |  6 ++++--
95   2 files changed, 36 insertions(+), 4 deletions(-)
96
97  commit 3f1aed640fc6fee3bf0994b069e662a528a5d4c8
98  Author: HongfeiChenCU <hc3222@columbia.edu>
99  Date:   Sun Apr 25 11:49:58 2021 -0700
100
101     half working array length
102
103  codegen.ml                   |  10 +++++++---
104  dev_tests/ast-test-arrays.jl |   2 +-
105  semant.ml                    |  43 ++++++++++++++++++++++++++++++-----------
106  stringfuncs.o                | Bin 3272 -> 0 bytes
107  tests/fail-array-length.err  |   1 +
108  tests/fail-array-length.jl   |   3 +++
109  tests/test-array-length1.jl  |   3 +++
110  tests/test-array-length1.out |   1 +
111  tests/test-array-length2.jl  |   9 +++++++++
112  tests/test-array-length2.out |   3 +++
113  tests/test-array1.jl         |   2 +-
114  tests/test-array2.jl         |   2 +-
115  tests/test-bubble-sort.jl    |   2 +-
116  tests/test-counting-sort.jl  |   4 ++--
117  14 files changed, 65 insertions(+), 20 deletions(-)
118
119 commit 9710da0c129c5e42a96817fe8801a9be29b81254
```

```
Author: Frances Cao <frances.te.cao@gmail.com>
Date:   Sun Apr 25 12:56:58 2021 -0400

    update class demo, add print pass and fail test, add buildandrun.log to
    ↪ makefile

 Makefile                                 |   4 +--
 demo/demo-class.jl                       |  48 ++++++++++++++++++++++++++++++
 demo/demo-class.out                      |   9 ++++++
 dev_tests/demo/demo-class.jl             |  48 ------------------------------
 dev_tests/demo/demo-class.out            |   8 -----
 dev_tests/stest-var.jl                   |   8 ++---
 stringfuncs.o                            | Bin 0 -> 3272 bytes
 tests/{fail-print.err => fail-print1.err} |   0
 tests/{fail-print.jl => fail-print1.jl}  |   0
 tests/fail-print2.err                    |   1 +
 tests/fail-print2.jl                     |   2 ++
 tests/test-print.jl                      |  29 ++++++++++++++++++
 tests/test-print.out                     |  13 ++++++++
 13 files changed, 107 insertions(+), 63 deletions(-)

commit 4b9acb6777d69e1cbfbc6ac238703d4abe07fd36
Author: Frances Cao <frances.te.cao@gmail.com>
Date:   Sun Apr 25 00:01:17 2021 -0400

    add print tests to dev

 dev_tests/demo/demo-class.jl  | 48 ++++++++++++++++++++++++++++++++++++++++++
 dev_tests/demo/demo-class.out |  8 ++++++++
 dev_tests/test-print.jl       | 29 ++++++++++++++++++++++++++
 tests/test-class5.jl          |  1 -
 4 files changed, 85 insertions(+), 1 deletion(-)

commit 3a7254364b618858cf3f265380676a2b609e7378
Merge: bdf0348 314bb3b
Author: Frances Cao <frances.te.cao@gmail.com>
Date:   Sat Apr 24 18:08:30 2021 -0400

    Merge branch 'main' of github.com:FranCao/javalite into main

commit bdf034830e88b018f5c30a6e2cdfc78d05e7cc75
Author: Frances Cao <frances.te.cao@gmail.com>
Date:   Sat Apr 24 18:08:21 2021 -0400

    cleanup external array funcs

 Makefile                   |   6 +----
 _tags                      |   5 +---
 arrayfuncs.c               |  61 ----------------------------------------------
 arrayfuncs.o               | Bin 1240 -> 0 bytes
 codegen.ml                 |   8 +++---
 testall.sh                 |  14 +++++-----
 tests/test-array-func1.jl  |   3 ---
 tests/test-array-func1.out |   1 -
 8 files changed, 13 insertions(+), 85 deletions(-)

commit 314bb3b5be3efc3c1007141bf5c57ca1e4ac51b6
```

```
Merge: 8ceea35 ba5c9ad
Author: HongfeiChenCU <hc3222@columbia.edu>
Date:   Sat Apr 24 14:34:06 2021 -0700

    Merge branch 'main' of https://github.com/FranCao/javalite into main

commit 8ceea3513afdac2f06290fb533ceaf6e14680c4c
Author: HongfeiChenCU <hc3222@columbia.edu>
Date:   Sat Apr 24 14:33:29 2021 -0700

    codegen warning resolved

 arrayfuncs.o  | Bin 1240 -> 0 bytes
 codegen.ml    |   2 +-
 stringfuncs.o | Bin 3272 -> 0 bytes
 3 files changed, 1 insertion(+), 1 deletion(-)

commit ba5c9ad1f42ff259b952aca3c7d9335e01f108be
Author: Mateo Maturana <mateomaturana@yahoo.com>
Date:   Sat Apr 24 17:05:21 2021 -0400

    test reorganization

 tests/fail-func5.err                   | 1 +
 tests/{fail-func9.jl => fail-func5.jl} | 2 +-
 tests/fail-func6.err                   | 2 +-
 tests/fail-func6.jl                    | 2 +-
 tests/fail-func9.err                   | 1 -
 5 files changed, 4 insertions(+), 4 deletions(-)

commit 50532e1daf6277db32c053f8fe88c4f5e35d1685
Author: Frances Cao <frances.te.cao@gmail.com>
Date:   Sat Apr 24 16:21:40 2021 -0400

    update boolean back to bool, cleanup unnecessary files

 arrayfuncs.o  | Bin 0 -> 1240 bytes
 ast.ml        |   2 +-
 gitlog.txt    | 510 ------------------------------------------------------------
 scanner.mll   |   2 +-
 stringfuncs.o | Bin 0 -> 3272 bytes
 5 files changed, 2 insertions(+), 512 deletions(-)

commit cb0876fdac7280d36c6479abf4e637301b3ac81a
Author: 314pies <ian121363@gmail.com>
Date:   Sat Apr 24 12:11:53 2021 -0700

    Add Java performence benchmarking code for compare

 tests/performance/Java/Person.java               | 11 ++++++++++
 tests/performance/Java/TestArrayAccess.java      | 20 ++++++++++++++++++++
 tests/performance/Java/TestClassFieldAccess.java | 11 ++++++++++
 tests/performance/Java/TestDoubleArithmetic.java | 20 ++++++++++++++++++++
 tests/performance/Java/TestIfCondition.java      | 16 ++++++++++++++++
 tests/performance/Java/TestIntArithmetic.java    | 20 ++++++++++++++++++++
 6 files changed, 98 insertions(+)

```

```
commit 63e9e6d880a142e11eac8476d47ab9c6446cb0a6
Merge: 2b2107c cadf4dc
Author: 314pies <ian121363@gmail.com>
Date:   Sat Apr 24 14:34:26 2021 -0400

    Merge branch 'main' into main

commit 2b2107c242996709c93bd90de37c95b57c875827
Author: 314pies <ian121363@gmail.com>
Date:   Sat Apr 24 11:28:27 2021 -0700

    Add Performance Benchmark

 PerformanceBench.sh                          |  19 ++++++++++++++++++
 arrayfuncs.o                                 | Bin 1240 -> 0 bytes
 stringfuncs.o                                | Bin 3272 -> 0 bytes
 tests/performance/test-Array-Access.jl       |  16 ++++++++++++++++
 tests/performance/test-Class-Field-Access.jl |  15 +++++++++++++++
 tests/performance/test-Double-Arithmetic.jl  |  18 ++++++++++++++++++
 tests/performance/test-If-Condition.jl       |  12 ++++++++++++
 tests/performance/test-Int-Arithmetic.jl     |  17 +++++++++++++++++
 8 files changed, 97 insertions(+)

commit 60c8a136b9243a68d03e88bdfc94c2e826e2d18f
Author: 314pies <ian121363@gmail.com>
Date:   Sat Apr 24 13:08:18 2021 -0400

    Add config.yml for CD/CI

 .circleci/config.yml | 14 ++++++++++++++
 1 file changed, 14 insertions(+)

commit cadf4dcb6460e72d6bd480db39044f155fb3c1d4
Merge: dbfc3fa 2573484
Author: 314pies <ian121363@gmail.com>
Date:   Sat Apr 24 12:52:27 2021 -0400

    Merge pull request #1 from 314pies/circleci-project-setup

    Add .circleci/config.yml

commit 2573484b131c5e332eb588b63ae391071f21fda2
Author: 314pies <ian121363@gmail.com>
Date:   Sat Apr 24 12:48:43 2021 -0400

    Add .circleci/config.yml

 .circleci/config.yml | 14 ++++++++++++++
 1 file changed, 14 insertions(+)

commit dbfc3faddd520cec6ef5d8e8c7d0bcd618191f2d
Author: Mateo Maturana <mateomaturana@yahoo.com>
Date:   Sat Apr 24 12:21:20 2021 -0400

    updates to git log

 gitlog.txt | 6 ++++++
```

```
 1 file changed, 6 insertions(+)

commit 87a730f264bdf03a968237282f3d6367535e2843
Author: Mateo Maturana <mateomaturana@yahoo.com>
Date:   Sat Apr 24 12:20:58 2021 -0400

    changes to test file names, added the git log

 gitlog.txt                              | 504 ++++++++++++++++++++++++++++++++++
 tests/{test-func9.jl => test-func7.jl}  |   2 +-
 tests/{test-func9.out => test-func7.out} |   0
 tests/test-func8.jl                     |   2 +-
 tests/{test-gcd.jl => test-gcd1.jl}     |   0
 tests/{test-gcd.out => test-gcd1.out}   |   0
 tests/{test-var3.jl => test-var2.jl}    |   0
 tests/{test-var3.out => test-var2.out}  |   0
 8 files changed, 506 insertions(+), 2 deletions(-)

commit a7ff0c4284d99c82c901d0240714fbd7d4ce546d
Author: Mateo Maturana <mateomaturana@yahoo.com>
Date:   Sat Apr 24 11:25:40 2021 -0400

    update bool -> boolean

 ast.ml      | 2 +-
 scanner.mll | 2 +-
 2 files changed, 2 insertions(+), 2 deletions(-)

commit c95703154f3b2315f3ccb670bfb25193464ea451
Author: Frances Cao <frances.te.cao@gmail.com>
Date:   Sat Apr 24 00:24:35 2021 -0400

    added some print array stuff, change arr layout, update null

 Makefile                     |  2 +-
 arrayfuncs.c                 | 21 ++++++++++++++++++-
 ast.ml                       |  4 ++--
 codegen.ml                   | 27 ++++++++++++++++++++----
 dev_tests/ast-test-arrays.jl | 41 ++++++++++++++++++++++++++++++++++++
 parser.mly                   |  2 +-
 scanner.mll                  |  2 +-
 semant.ml                    | 11 +++++++++-
 tests/fail-array1.err        |  2 +-
 tests/test-class6.jl         | 16 ++++++++++++++
 10 files changed, 85 insertions(+), 43 deletions(-)

commit 5eea65c90d2a3f1c5914c1c6fc54a6b6e2e89760
Author: Frances Cao <frances.te.cao@gmail.com>
Date:   Thu Apr 22 17:39:09 2021 -0400

    pieces of length for array

 Makefile                   |   6 +++++-
 _tags                      |   5 ++++-
 arrayfuncs.c               |  42 ++++++++++++++++++++++++++++++++++++++++++
 arrayfuncs.o               | Bin 0 -> 1240 bytes
 codegen.ml                 |   8 ++++++++
```

```
 semant.ml                   |  13 +++++++++++++
 testall.sh                  |   9 ++++++++-
 tests/test-array-func1.jl   |   3 +++
 tests/test-array-func1.out  |   1 +
 tests/test-array2.jl        |   1 -
 tests/test-performance1.jl  |   8 --------
 tests/test-performance1.out |   1 -
 12 files changed, 84 insertions(+), 13 deletions(-)

commit cd8f383f357f7a0d4c2c638534385d2b3069a494
Author: FranCao <frances.te.cao@gmail.com>
Date:   Thu Apr 22 15:33:46 2021 -0400

    cleanup

 SampleCode1.jl | 3 ---
 1 file changed, 3 deletions(-)

commit 29e741924b0d487bd509f9ad244a9fe36ec6306c
Author: HongfeiChenCU <hc3222@columbia.edu>
Date:   Thu Apr 22 08:21:08 2021 +0000

    linkedlist test added

 codegen.ml                | 17 ++++++++++-------
 dev_tests/aatest-class.jl | 26 ++++++++++++++-----------
 dev_tests/classinc.c      |  5 +++--
 semant.ml                 | 27 +++++---------------------
 tests/test-class6.jl      | 48 +++++++++++++++++++++++++++++++++++++++++++++++
 tests/test-class6.out     |  8 ++++++++
 6 files changed, 89 insertions(+), 42 deletions(-)

commit 777a2e23114ddf6d98db61e2a8b60ae6fea6bfb4
Merge: cd6b094 f1e952e
Author: HongfeiChenCU <hc3222@columbia.edu>
Date:   Wed Apr 21 23:22:02 2021 -0700

    Merge branch 'main' into hc-inheritance

commit cd6b094715abaa03ecef7dfffa32145932620af3
Author: HongfeiChenCU <hc3222@columbia.edu>
Date:   Wed Apr 21 23:17:57 2021 -0700

    recursive class and null pointer done

 ast.ml                    |  5 +++-
 codegen.ml                | 26 ++++++++++++++-----
 dev_tests/aatest-class.jl | 23 +++++------------
 dev_tests/classinc.c      |  6 +++--
 parser.mly                |  2 ++
 sast.ml                   |  2 ++
 scanner.mll               |  2 ++
 semant.ml                 | 66 +++++++++++++++++++++++++++++++++++------------
 8 files changed, 90 insertions(+), 42 deletions(-)

commit f1e952e4b34cf0c8a8a48a7a224c436a3fa96700
Author: Mateo Maturana <mateomaturana@yahoo.com>
```

```
Date:   Wed Apr 21 18:02:20 2021 -0400

    added return check to main in semant

 semant.ml              | 6 ++++--
 tests/fail-for1.jl     | 5 ++++-
 tests/fail-return3.err | 1 +
 tests/fail-return3.jl  | 2 ++
 tests/test-add1.jl     | 1 -
 tests/test-while2.jl   | 1 -
 6 files changed, 11 insertions(+), 5 deletions(-)

commit 43d0873a90726dd93b54fb3a36153e01a3cf0c83
Author: 314pies <ian121363@gmail.com>
Date:   Wed Apr 21 14:19:50 2021 -0700

    remove test-array-func1, add fail-return1 test

 tests/fail-return1.err     | 1 +
 tests/fail-return1.jl      | 4 ++++
 tests/test-array-func1.jl  | 6 ------
 tests/test-array-func1.out | 1 -
 4 files changed, 5 insertions(+), 7 deletions(-)

commit b013dd6747aca6fe4205b6f90fde3fc0000d6a46
Merge: c150b0e a8d00d8
Author: 314pies <ian121363@gmail.com>
Date:   Wed Apr 21 16:54:17 2021 -0400

    Merge pull request #2 from FranCao/functional

    Functional

commit a8d00d8c5d14bff07a2c6a8c181ea4f52f38cd6e
Author: 314pies <ian121363@gmail.com>
Date:   Wed Apr 21 13:52:49 2021 -0700

    Fixes duplicate functional declaration in test cases

 SampleCode1.jl              | 10 +++------
 tests/fail-array1.jl        |  5 +----
 tests/fail-array2.jl        |  5 +----
 tests/fail-array3.jl        | 13 +++++------
 tests/fail-array4.jl        |  7 ++----
 tests/fail-assign1.jl       | 13 +++++------
 tests/fail-assign2.jl       |  8 +++----
 tests/fail-assign3.jl       |  7 +++---
 tests/fail-class1.jl        |  9 +++-----
 tests/fail-class2.jl        |  9 +++-----
 tests/fail-dead1.jl         |  9 +++-----
 tests/fail-dead2.jl         | 14 +++++-------
 tests/fail-double1.jl       |  6 +----
 tests/fail-double2.jl       |  6 +----
 tests/fail-expr1.jl         |  5 -----
 tests/fail-expr2.jl         |  5 -----
 tests/fail-expr3.jl         |  5 -----
 tests/fail-for1.jl          | 17 +++++-----------
```

```
tests/fail-for2.jl          |  9 ++------
tests/fail-for3.jl          | 10 ++-------
tests/fail-for4.jl          |  9 ++------
tests/fail-for5.jl          | 14 +++++-------
tests/fail-func1.jl         |  4 ----
tests/fail-func2.jl         |  4 ----
tests/fail-func3.jl         |  4 ----
tests/fail-func4.jl         |  4 ----
tests/fail-func6.jl         |  9 ++++----
tests/fail-func7.jl         |  9 ++++----
tests/fail-func8.jl         |  9 ++++----
tests/fail-func9.jl         |  9 ++++----
tests/fail-if1.jl           | 11 +++++-----
tests/fail-if2.jl           |  7 ++----
tests/fail-if3.jl           | 12 +++++-----
tests/fail-nomain.err       |  1 -
tests/fail-nomain.jl        |  0
tests/fail-return1.err      |  1 -
tests/fail-return1.jl       |  4 ----
tests/fail-return2.jl       |  5 -----
tests/fail-string-binop1.jl | 12 +++++-----
tests/fail-while1.jl        | 17 ++++++---------
tests/fail-while2.jl        | 16 +++++---------
tests/test-add1.jl          |  9 ++++----
tests/test-arith1.jl        |  6 +----
tests/test-arith2.jl        |  6 +----
tests/test-arith3.jl        | 10 +++------
tests/test-arith4.jl        |  6 +----
tests/test-array-func1.jl   | 12 +++++-----
tests/test-array1.jl        | 18 +++++++--------
tests/test-array2.jl        | 21 +++++++++---------
tests/test-array3.jl        | 14 +++++-------
tests/test-array4.jl        | 20 ++++++++---------
tests/test-array5.jl        | 21 +++++++++---------
tests/test-bubble-sort.jl   | 40 +++++++++++++++-------------------
tests/test-class1.jl        | 19 ++++++++--------
tests/test-class2.jl        |  8 +++----
tests/test-class3.jl        | 20 ++++++++---------
tests/test-class4.jl        | 10 ++++-----
tests/test-class5.jl        | 19 ++++++++--------
tests/test-counting-sort.jl | 34 +++++++++++++----------------
tests/test-double1.jl       |  8 ++-----
tests/test-double2.jl       | 13 +++++------
tests/test-double3.jl       | 12 ++++------
tests/test-fib.jl           | 16 +++++---------
tests/test-for1.jl          | 11 ++++------
tests/test-for2.jl          | 14 +++++-------
tests/test-func-string1.jl  | 24 ++++++++++----------
tests/test-func-string2.jl  | 18 +++++++--------
tests/test-func-string3.jl  | 29 ++++++++++++-------------
tests/test-func-string4.jl  | 32 +++++++++++++--------------
tests/test-func1.jl         |  9 +++-----
tests/test-func2.jl         | 11 +++-------
tests/test-func3.jl         |  7 ++----
tests/test-func4.jl         | 10 ++++-----
tests/test-func5.jl         |  5 -----
tests/test-func6.jl         |  8 +++----
```

```
 tests/test-func8.jl         |  8 +++----
 tests/test-func9.jl         |  7 ++----
 tests/test-gcd.jl           | 11 ++++------
 tests/test-gcd2.jl          | 10 +++------
 tests/test-if1.jl           |  8 ++-----
 tests/test-if2.jl           |  8 ++-----
 tests/test-if3.jl           |  8 ++-----
 tests/test-if4.jl           |  8 ++-----
 tests/test-if5.jl           | 11 +++++-----
 tests/test-if6.jl           | 10 ++++-----
 tests/test-local1.jl        |  7 ++----
 tests/test-local2.jl        |  5 +----
 tests/test-ops1.jl          | 53 +++++++++++++++++++++------------------------------------
 tests/test-ops2.jl          | 32 +++++++++++++--------------
 tests/test-performance1.jl  | 18 ++++++---------
 tests/test-string-binop1.jl | 11 ++++------
 tests/test-string-binop2.jl | 22 +++++++++----------
 tests/test-var1.jl          |  8 ++-----
 tests/test-var3.jl          | 11 ++++------
 tests/test-while1.jl        | 14 +++++-------
 tests/test-while2.jl        |  9 ++++----
 96 files changed, 431 insertions(+), 701 deletions(-)

commit 86a733aaa2f1c4146c23b161b146f10ee5ddc392
Author: Mateo Maturana <mateomaturana@yahoo.com>
Date:   Wed Apr 21 15:56:49 2021 -0400

    main fix done -- test cases need to be changed

 ast.ml                      |  7 ++++---
 parser.mly                  | 20 ++++++++++++--------
 semant.ml                   | 12 +++++++++++-
 tests/fail-functional1.err |  1 +
 tests/fail-functional1.jl  |  9 +++++++++
 tests/test-functional1.jl  |  2 ++
 tests/test-functional1.out |  1 +
 tests/test-functional2.jl  | 11 +++++++++++
 tests/test-functional2.out |  2 ++
 tests/test-functional3.jl  | 20 ++++++++++++++++++++
 tests/test-functional3.out |  4 ++++
 11 files changed, 77 insertions(+), 12 deletions(-)

commit c150b0e5ecf8c66883c64889ed4890ab4e1ef7fb
Merge: d13682f 118a270
Author: Mateo Maturana <mateomaturana@yahoo.com>
Date:   Wed Apr 21 13:08:40 2021 -0400

    Merge branch 'main' of github.com:FranCao/javalite into main

commit d13682fb5fbc96ba21b0bc1a07eb236fbe6a3368
Author: Mateo Maturana <mateomaturana@yahoo.com>
Date:   Wed Apr 21 13:08:34 2021 -0400

    minor change to test-for1

 tests/test-for1.jl | 3 +--
 1 file changed, 1 insertion(+), 2 deletions(-)
```

```
575
576 commit 118a270094844afc3ab53aa8f1a887a4bfa59c98
577 Author: HongfeiChenCU <hc3222@columbia.edu>
578 Date:   Tue Apr 20 22:10:58 2021 -0700
579
580     local hide formal updated
581
582  codegen.ml           |  7 ++++++--
583  semant.ml            | 13 ++++++++------
584  tests/test-local2.jl |  2 +-
585  3 files changed, 14 insertions(+), 8 deletions(-)
586
587 commit 6b8db289e4fce10398d642f0873057c6e359b783
588 Author: Mateo Maturana <mateomaturana@yahoo.com>
589 Date:   Wed Apr 21 01:02:05 2021 -0400
590
591     fix tests, remove some, rename
592
593  tests/fail-assign1.jl                                | 3 +--
594  tests/fail-assign2.err                               | 2 +-
595  tests/fail-assign3.err                               | 2 +-
596  tests/fail-dead1.jl                                  | 1 +
597  tests/test-arith5.jl                                 | 5 -----
598  tests/test-arith5.out                                | 1 -
599  tests/test-arith6.jl                                 | 5 -----
600  tests/test-arith6.out                                | 1 -
601  tests/test-counting-sort.jl                          | 4 ++--
602  tests/{test-performence1.jl => test-performance1.jl}   | 0
603  tests/{test-performence1.out => test-performance1.out} | 0
604  11 files changed, 6 insertions(+), 18 deletions(-)
605
606 commit c45a9c3bec03fae95f77c5fc6f96b62e1a022198
607 Author: Mateo Maturana <mateomaturana@yahoo.com>
608 Date:   Tue Apr 20 23:26:07 2021 -0400
609
610     refactoring/fixing of tests + new arith tests
611
612  Makefile                                 |   2 +-
613  stringfuncs.o                            | Bin 0 -> 3272 bytes
614  {testsold => tests}/fail-assign1.err     |   0
615  {testsold => tests}/fail-assign1.jl      |   7 ++-----
616  {testsold => tests}/fail-assign2.err     |   0
617  tests/fail-assign2.jl                    |   5 +++++
618  {testsold => tests}/fail-assign3.err     |   0
619  tests/fail-assign3.jl                    |   9 +++++++++
620  {testsold => tests}/fail-dead1.err       |   0
621  {testsold => tests}/fail-dead1.jl        |   5 +----
622  {testsold => tests}/fail-dead2.err       |   0
623  {testsold => tests}/fail-dead2.jl        |   2 +-
624  {testsold => tests}/fail-double1.err     |   0
625  {testsold => tests}/fail-double1.jl      |   0
626  {testsold => tests}/fail-double2.err     |   0
627  {testsold => tests}/fail-double2.jl      |   0
628  {testsold => tests}/fail-expr1.err       |   2 +-
629  tests/fail-expr1.jl                      |  11 ++++++++++
630  {testsold => tests}/fail-expr2.err       |   2 +-
631  tests/fail-expr2.jl                      |  11 ++++++++++
```

```
 {testsold => tests}/fail-expr3.err         |   2 +-
 tests/fail-expr3.jl                        |  11 ++++++++++
 {testsold => tests}/fail-for1.err          |   0
 {testsold => tests}/fail-for1.jl           |   2 +-
 {testsold => tests}/fail-for2.err          |   0
 {testsold => tests}/fail-for2.jl           |   2 +-
 {testsold => tests}/fail-for3.err          |   0
 {testsold => tests}/fail-for3.jl           |   2 +-
 {testsold => tests}/fail-for4.err          |   0
 {testsold => tests}/fail-for4.jl           |   2 +-
 {testsold => tests}/fail-for5.err          |   0
 {testsold => tests}/fail-for5.jl           |   2 +-
 {testsold => tests}/fail-func1.err         |   0
 {testsold => tests}/fail-func1.jl          |   0
 {testsold => tests}/fail-func2.err         |   0
 {testsold => tests}/fail-func2.jl          |   0
 {testsold => tests}/fail-func3.err         |   0
 {testsold => tests}/fail-func3.jl          |   0
 {testsold => tests}/fail-func4.err         |   0
 {testsold => tests}/fail-func4.jl          |   0
 {testsold => tests}/fail-func6.err         |   0
 {testsold => tests}/fail-func6.jl          |   0
 {testsold => tests}/fail-func7.err         |   0
 {testsold => tests}/fail-func7.jl          |   0
 {testsold => tests}/fail-func8.err         |   0
 {testsold => tests}/fail-func8.jl          |   0
 {testsold => tests}/fail-func9.err         |   0
 {testsold => tests}/fail-func9.jl          |   0
 {testsold => tests}/fail-if1.err           |   0
 {testsold => tests}/fail-if1.jl            |   0
 {testsold => tests}/fail-if2.err           |   0
 {testsold => tests}/fail-if2.jl            |   0
 {testsold => tests}/fail-if3.err           |   0
 {testsold => tests}/fail-if3.jl            |   0
 {testsold => tests}/fail-nomain.err        |   0
 {testsold => tests}/fail-nomain.jl         |   0
 {testsold => tests}/fail-print.err         |   0
 {testsold => tests}/fail-print.jl          |   0
 {testsold => tests}/fail-return1.err       |   0
 {testsold => tests}/fail-return1.jl        |   0
 {testsold => tests}/fail-return2.err       |   0
 {testsold => tests}/fail-return2.jl        |   0
 {testsold => tests}/fail-while1.err        |   0
 {testsold => tests}/fail-while1.jl         |   2 +-
 {testsold => tests}/fail-while2.err        |   0
 {testsold => tests}/fail-while2.jl         |   2 +-
 {testsold => tests}/test-add1.jl           |   0
 {testsold => tests}/test-add1.out          |   0
 {testsold => tests}/test-arith1.jl         |   0
 {testsold => tests}/test-arith1.out        |   0
 {testsold => tests}/test-arith2.jl         |   0
 {testsold => tests}/test-arith2.out        |   0
 {testsold => tests}/test-arith3.jl         |   3 +--
 {testsold => tests}/test-arith3.out        |   0
 tests/test-arith4.jl                       |   5 +++++
 tests/test-arith4.out                      |   1 +
 tests/test-arith5.jl                       |   5 +++++
```

```
tests/test-arith5.out                       |   1 +
tests/test-arith6.jl                        |   5 +++++
tests/test-arith6.out                       |   1 +
tests/test-bubble-sort.jl                   |  25 ++++++++++++++++++++++++
{testsold => tests}/test-bubble-sort.out    |   0
{testsold => tests}/test-counting-sort.jl   |  19 +++++++-----------
{testsold => tests}/test-counting-sort.out  |   0
{testsold => tests}/test-double1.jl         |   3 +--
{testsold => tests}/test-double1.out        |   0
tests/test-double2.jl                       |   8 ++++++++
{testsold => tests}/test-double2.out        |   0
{testsold => tests}/test-double3.jl         |   7 ++-----
{testsold => tests}/test-double3.out        |   0
{testsold => tests}/test-fib.jl             |   0
{testsold => tests}/test-fib.out            |   0
{testsold => tests}/test-for1.jl            |   2 +-
{testsold => tests}/test-for1.out           |   0
{testsold => tests}/test-for2.jl            |   3 +--
{testsold => tests}/test-for2.out           |   0
{testsold => tests}/test-func1.jl           |   3 +--
{testsold => tests}/test-func1.out          |   0
{testsold => tests}/test-func2.jl           |   3 +--
{testsold => tests}/test-func2.out          |   0
{testsold => tests}/test-func3.jl           |   0
{testsold => tests}/test-func3.out          |   0
{testsold => tests}/test-func4.jl           |   6 ++----
{testsold => tests}/test-func4.out          |   0
{testsold => tests}/test-func5.jl           |   0
{testsold => tests}/test-func5.out          |   0
{testsold => tests}/test-func6.jl           |   0
{testsold => tests}/test-func6.out          |   0
{testsold => tests}/test-func8.jl           |   0
{testsold => tests}/test-func8.out          |   0
{testsold => tests}/test-func9.jl           |   0
{testsold => tests}/test-func9.out          |   0
{testsold => tests}/test-gcd.jl             |   0
{testsold => tests}/test-gcd.out            |   0
{testsold => tests}/test-gcd2.jl            |   0
{testsold => tests}/test-gcd2.out           |   0
{testsold => tests}/test-if1.jl             |   0
{testsold => tests}/test-if1.out            |   0
{testsold => tests}/test-if2.jl             |   0
{testsold => tests}/test-if2.out            |   0
{testsold => tests}/test-if3.jl             |   0
{testsold => tests}/test-if3.out            |   0
{testsold => tests}/test-if4.jl             |   0
{testsold => tests}/test-if4.out            |   0
{testsold => tests}/test-if5.jl             |   2 +-
{testsold => tests}/test-if5.out            |   0
{testsold => tests}/test-if6.jl             |   3 +--
{testsold => tests}/test-if6.out            |   0
{testsold => tests}/test-local1.jl          |   4 +---
{testsold => tests}/test-local1.out         |   0
{testsold => tests}/test-local2.jl          |   7 ++-----
{testsold => tests}/test-local2.out         |   0
{testsold => tests}/test-ops1.jl            |   0
{testsold => tests}/test-ops1.out           |   0
```

```
 {testsold => tests}/test-ops2.jl           |   0
 {testsold => tests}/test-ops2.out          |   0
 {testsold => tests}/test-performence1.jl   |   6 ++----
 {testsold => tests}/test-performence1.out  |   0
 {testsold => tests}/test-var1.jl           |   3 +--
 {testsold => tests}/test-var1.out          |   0
 tests/test-var3.jl                         |   8 ++++++++
 {testsold => tests}/test-var3.out          |   0
 {testsold => tests}/test-while1.jl         |   3 +--
 {testsold => tests}/test-while1.out        |   0
 {testsold => tests}/test-while2.jl         |   3 +--
 {testsold => tests}/test-while2.out        |   0
 testsold/fail-assign2.jl                   |   7 -------
 testsold/fail-assign3.jl                   |  11 -----------
 testsold/fail-expr1.jl                     |  18 ------------------
 testsold/fail-expr2.jl                     |  14 --------------
 testsold/fail-expr3.jl                     |  14 --------------
 testsold/fail-func5.err                    |   1 -
 testsold/fail-func5.jl                     |  14 --------------
 testsold/fail-global1.err                  |   1 -
 testsold/fail-global1.jl                   |   9 ---------
 testsold/fail-global2.err                  |   1 -
 testsold/fail-global2.jl                   |   9 ---------
 testsold/test-aglobal4.jl                  |   7 -------
 testsold/test-aglobal4.out                 |   2 --
 testsold/test-bubble-sort.jl               |  27 ---------------------------
 testsold/test-double2.jl                   |  11 -----------
 testsold/test-func7.jl                     |  13 -------------
 testsold/test-func7.out                    |   1 -
 testsold/test-global1.jl                   |  30 ------------------------------
 testsold/test-global1.out                  |   4 ----
 testsold/test-global2.jl                   |  10 ----------
 testsold/test-global2.out                  |   1 -
 testsold/test-global3.jl                   |  11 -----------
 testsold/test-global3.out                  |   1 -
 testsold/test-var2.jl                      |  13 -------------
 testsold/test-var2.out                     |   1 -
 testsold/test-var3.jl                      |  10 ----------
 172 files changed, 148 insertions(+), 315 deletions(-)

commit 09308ae1f1aa31e422bc3eb5ca506c7f80c06b3c
Author: Mateo Maturana <mateomaturana@yahoo.com>
Date:   Tue Apr 20 22:07:30 2021 -0400

    fixed Makefile, removed stringfuncs.o

 Makefile      |   4 ++--
 stringfuncs.o | Bin 3272 -> 0 bytes
 2 files changed, 2 insertions(+), 2 deletions(-)

commit bc4d028dc5d333f8bc5d43b0e85031851d5f91d3
Author: Mateo Maturana <mateomaturana@yahoo.com>
Date:   Tue Apr 20 22:06:26 2021 -0400

    added object files to clean target Makefile

 Makefile | 2 +-
```

```
 1 file changed, 1 insertion(+), 1 deletion(-)

commit 26c8415223f1944e42a10501dfb1f33c46a4a6e1
Author: Frances Cao <frances.te.cao@gmail.com>
Date:   Tue Apr 20 19:09:06 2021 -0400

    update tests, cleanup old files

 funcs/reversestring.c                          | 27 --------------------
 funcs/stringlower.c                            | 28 --------------------
 funcs/stringsubstring.c                        | 31 ----------------------
 funcs/stringupper.c                            | 28 --------------------
 tests/fail-string-binop1.err                   |  1 +
 tests/fail-string-binop1.jl                    |  6 +++++
 {testsold => tests}/test-array-func1.jl        |  0
 {testsold => tests}/test-array-func1.out       |  0
 tests/test-func-string1.jl                     | 14 ++++++++++
 tests/test-func-string1.out                    |  5 ++++
 tests/test-func-string2.jl                     | 10 +++++++
 .../test-func-string2.out                      |  0
 .../test-func-string3.jl                       | 11 ++++++----
 .../test-func-string3.out                      |  0
 tests/test-func-string4.jl                     | 22 ++++++++++++++++
 .../test-func-string4.out                      |  0
 tests/test-string-binop1.jl                    |  8 ++++++
 tests/test-string-binop1.out                   |  2 ++
 testsold/test-afunc-string1.jl                 | 11 --------
 testsold/test-afunc-string1.out                |  3 ---
 testsold/test-afunc-string2.jl                 |  9 -------
 testsold/test-afunc-string2.out                |  2 --
 testsold/test-afunc-string3.jl                 | 14 ----------
 testsold/test-afunc-string5.jl                 | 29 --------------------
 testsold/test-astring-binop.jl                 |  6 -----
 testsold/test-astring-binop.out                |  1 -
 testsold/test-hello.jl                         |  7 -----
 testsold/test-hello.out                        |  3 ---
 testsold/test-helloworld.jl                    |  4 ---
 testsold/test-helloworld.out                   |  1 -
 testsold/test-printstring.jl                   |  6 -----
 testsold/test-printstring.out                  |  1 -
 32 files changed, 73 insertions(+), 217 deletions(-)

commit 7a6278e9facc35d4ea3ac188b42b140c2283f4f7
Author: HongfeiChenCU <hc3222@columbia.edu>
Date:   Tue Apr 20 13:49:49 2021 -0700

    variable decl assign done; tests partially updated

 codegen.ml                                     | 56 ++++++------------------------
 semant.ml                                      | 14 ++++++----
 tests/fail-array1.err                          |  2 +-
 tests/fail-array1.jl                           |  3 +-
 tests/fail-array2.jl                           |  3 +-
 tests/fail-array3.jl                           | 11 +++----
 tests/fail-array4.jl                           |  3 +-
 tests/fail-class1.jl                           |  9 ++---
 tests/fail-class2.err                          |  2 +-
```

```
tests/fail-class2.jl                      |  9 ++---
tests/test-array1.jl                      |  9 ++---
tests/test-array2.jl                      |  8 ++---
tests/test-array3.jl                      |  9 ++---
tests/test-array4.jl                      |  6 ++--
tests/test-array5.jl                      |  6 ++--
tests/test-class3.jl                      |  3 +-
tests/test-class5.jl                      | 43 ++++++----------------
tests/test-string-binop2.jl               |  9 ++---
{tests => testsold}/fail-assign1.err      |  0
{tests => testsold}/fail-assign1.jl       |  0
{tests => testsold}/fail-assign2.err      |  0
{tests => testsold}/fail-assign2.jl       |  0
{tests => testsold}/fail-assign3.err      |  0
{tests => testsold}/fail-assign3.jl       |  0
{tests => testsold}/fail-dead1.err        |  0
{tests => testsold}/fail-dead1.jl         |  0
{tests => testsold}/fail-dead2.err        |  0
{tests => testsold}/fail-dead2.jl         |  0
{tests => testsold}/fail-double1.err      |  0
{tests => testsold}/fail-double1.jl       |  0
{tests => testsold}/fail-double2.err      |  0
{tests => testsold}/fail-double2.jl       |  0
{tests => testsold}/fail-expr1.err        |  0
{tests => testsold}/fail-expr1.jl         |  0
{tests => testsold}/fail-expr2.err        |  0
{tests => testsold}/fail-expr2.jl         |  0
{tests => testsold}/fail-expr3.err        |  0
{tests => testsold}/fail-expr3.jl         |  0
{tests => testsold}/fail-for1.err         |  0
{tests => testsold}/fail-for1.jl          |  0
{tests => testsold}/fail-for2.err         |  0
{tests => testsold}/fail-for2.jl          |  0
{tests => testsold}/fail-for3.err         |  0
{tests => testsold}/fail-for3.jl          |  0
{tests => testsold}/fail-for4.err         |  0
{tests => testsold}/fail-for4.jl          |  0
{tests => testsold}/fail-for5.err         |  0
{tests => testsold}/fail-for5.jl          |  0
{tests => testsold}/fail-func1.err        |  0
{tests => testsold}/fail-func1.jl         |  0
{tests => testsold}/fail-func2.err        |  0
{tests => testsold}/fail-func2.jl         |  0
{tests => testsold}/fail-func3.err        |  0
{tests => testsold}/fail-func3.jl         |  0
{tests => testsold}/fail-func4.err        |  0
{tests => testsold}/fail-func4.jl         |  0
{tests => testsold}/fail-func5.err        |  0
{tests => testsold}/fail-func5.jl         |  0
{tests => testsold}/fail-func6.err        |  0
{tests => testsold}/fail-func6.jl         |  0
{tests => testsold}/fail-func7.err        |  0
{tests => testsold}/fail-func7.jl         |  0
{tests => testsold}/fail-func8.err        |  0
{tests => testsold}/fail-func8.jl         |  0
{tests => testsold}/fail-func9.err        |  0
{tests => testsold}/fail-func9.jl         |  0
```

```
{tests => testsold}/fail-global1.err       |  0
{tests => testsold}/fail-global1.jl        |  0
{tests => testsold}/fail-global2.err       |  0
{tests => testsold}/fail-global2.jl        |  0
{tests => testsold}/fail-if1.err           |  0
{tests => testsold}/fail-if1.jl            |  0
{tests => testsold}/fail-if2.err           |  0
{tests => testsold}/fail-if2.jl            |  0
{tests => testsold}/fail-if3.err           |  0
{tests => testsold}/fail-if3.jl            |  0
{tests => testsold}/fail-nomain.err        |  0
{tests => testsold}/fail-nomain.jl         |  0
{tests => testsold}/fail-print.err         |  0
{tests => testsold}/fail-print.jl          |  0
{tests => testsold}/fail-return1.err       |  0
{tests => testsold}/fail-return1.jl        |  0
{tests => testsold}/fail-return2.err       |  0
{tests => testsold}/fail-return2.jl        |  0
{tests => testsold}/fail-while1.err        |  0
{tests => testsold}/fail-while1.jl         |  0
{tests => testsold}/fail-while2.err        |  0
{tests => testsold}/fail-while2.jl         |  0
{tests => testsold}/test-add1.jl           |  0
{tests => testsold}/test-add1.out          |  0
{tests => testsold}/test-afunc-string1.jl  |  0
{tests => testsold}/test-afunc-string1.out |  0
{tests => testsold}/test-afunc-string2.jl  |  0
{tests => testsold}/test-afunc-string2.out |  0
{tests => testsold}/test-afunc-string3.jl  |  0
{tests => testsold}/test-afunc-string3.out |  0
{tests => testsold}/test-afunc-string4.jl  |  0
{tests => testsold}/test-afunc-string4.out |  0
{tests => testsold}/test-afunc-string5.jl  |  0
{tests => testsold}/test-afunc-string5.out |  0
{tests => testsold}/test-aglobal4.jl       |  0
{tests => testsold}/test-aglobal4.out      |  0
{tests => testsold}/test-arith1.jl         |  0
{tests => testsold}/test-arith1.out        |  0
{tests => testsold}/test-arith2.jl         |  0
{tests => testsold}/test-arith2.out        |  0
{tests => testsold}/test-arith3.jl         |  0
{tests => testsold}/test-arith3.out        |  0
{tests => testsold}/test-array-func1.jl    |  0
{tests => testsold}/test-array-func1.out   |  0
{tests => testsold}/test-astring-binop.jl  |  0
{tests => testsold}/test-astring-binop.out |  0
{tests => testsold}/test-bubble-sort.jl    |  0
{tests => testsold}/test-bubble-sort.out   |  0
{tests => testsold}/test-counting-sort.jl  |  0
{tests => testsold}/test-counting-sort.out |  0
{tests => testsold}/test-double1.jl        |  0
{tests => testsold}/test-double1.out       |  0
{tests => testsold}/test-double2.jl        |  0
{tests => testsold}/test-double2.out       |  0
{tests => testsold}/test-double3.jl        |  0
{tests => testsold}/test-double3.out       |  0
{tests => testsold}/test-fib.jl            |  0
```

```
{tests => testsold}/test-fib.out           |  0
{tests => testsold}/test-for1.jl           |  0
{tests => testsold}/test-for1.out          |  0
{tests => testsold}/test-for2.jl           |  0
{tests => testsold}/test-for2.out          |  0
{tests => testsold}/test-func1.jl          |  0
{tests => testsold}/test-func1.out         |  0
{tests => testsold}/test-func2.jl          |  0
{tests => testsold}/test-func2.out         |  0
{tests => testsold}/test-func3.jl          |  0
{tests => testsold}/test-func3.out         |  0
{tests => testsold}/test-func4.jl          |  0
{tests => testsold}/test-func4.out         |  0
{tests => testsold}/test-func5.jl          |  0
{tests => testsold}/test-func5.out         |  0
{tests => testsold}/test-func6.jl          |  0
{tests => testsold}/test-func6.out         |  0
{tests => testsold}/test-func7.jl          |  0
{tests => testsold}/test-func7.out         |  0
{tests => testsold}/test-func8.jl          |  0
{tests => testsold}/test-func8.out         |  0
{tests => testsold}/test-func9.jl          |  0
{tests => testsold}/test-func9.out         |  0
{tests => testsold}/test-gcd.jl            |  0
{tests => testsold}/test-gcd.out           |  0
{tests => testsold}/test-gcd2.jl           |  0
{tests => testsold}/test-gcd2.out          |  0
{tests => testsold}/test-global1.jl        |  0
{tests => testsold}/test-global1.out       |  0
{tests => testsold}/test-global2.jl        |  0
{tests => testsold}/test-global2.out       |  0
{tests => testsold}/test-global3.jl        |  0
{tests => testsold}/test-global3.out       |  0
{tests => testsold}/test-hello.jl          |  0
{tests => testsold}/test-hello.out         |  0
{tests => testsold}/test-helloworld.jl     |  0
{tests => testsold}/test-helloworld.out    |  0
{tests => testsold}/test-if1.jl            |  0
{tests => testsold}/test-if1.out           |  0
{tests => testsold}/test-if2.jl            |  0
{tests => testsold}/test-if2.out           |  0
{tests => testsold}/test-if3.jl            |  0
{tests => testsold}/test-if3.out           |  0
{tests => testsold}/test-if4.jl            |  0
{tests => testsold}/test-if4.out           |  0
{tests => testsold}/test-if5.jl            |  0
{tests => testsold}/test-if5.out           |  0
{tests => testsold}/test-if6.jl            |  0
{tests => testsold}/test-if6.out           |  0
{tests => testsold}/test-local1.jl         |  0
{tests => testsold}/test-local1.out        |  0
{tests => testsold}/test-local2.jl         |  0
{tests => testsold}/test-local2.out        |  0
{tests => testsold}/test-ops1.jl           |  0
{tests => testsold}/test-ops1.out          |  0
{tests => testsold}/test-ops2.jl           |  0
{tests => testsold}/test-ops2.out          |  0
```

```
 {tests => testsold}/test-performence1.jl   |  0
 {tests => testsold}/test-performence1.out  |  0
 {tests => testsold}/test-printstring.jl    |  0
 {tests => testsold}/test-printstring.out   |  0
 {tests => testsold}/test-var1.jl           |  0
 {tests => testsold}/test-var1.out          |  0
 {tests => testsold}/test-var2.jl           |  0
 {tests => testsold}/test-var2.out          |  0
 {tests => testsold}/test-var3.jl           |  0
 {tests => testsold}/test-var3.out          |  0
 {tests => testsold}/test-while1.jl         |  0
 {tests => testsold}/test-while1.out        |  0
 {tests => testsold}/test-while2.jl         |  0
 {tests => testsold}/test-while2.out        |  0
 194 files changed, 59 insertions(+), 146 deletions(-)

commit 60f2abfb52f0f2a26ff62777e2bd63a7cd97bfbb
Author: HongfeiChenCU <hc3222@columbia.edu>
Date:   Tue Apr 20 12:23:15 2021 -0700

    still in progress

 codegen.ml             | 42 ++++++++++++++++++++++++++++++++--------
 dev_tests/stest-var.jl |  3 ++-
 parser.mly             |  2 +-
 semant.ml              | 17 ++++++++++-------
 tests/test-class1.jl   |  3 +--
 tests/test-class2.jl   | 12 ++++++------
 tests/test-class3.jl   | 27 +++++++++++++++++++--------
 7 files changed, 61 insertions(+), 45 deletions(-)

commit 9d4f97f0d3a612e1d1a231d724d90d3fa27906ec
Author: HongfeiChenCU <hc3222@columbia.edu>
Date:   Tue Apr 20 10:39:15 2021 -0700

    up to semant (remove extra)

 _build/_digests            |   32 -
 _build/_log                |   80 ---
 _build/ast.cmi             |  Bin 3328 -> 0 bytes
 _build/ast.cmo             |  Bin 5857 -> 0 bytes
 _build/ast.cmx             |  Bin 687 -> 0 bytes
 _build/ast.ml              |  139 ----
 _build/ast.ml.depends      |    1 -
 _build/ast.o               |  Bin 25760 -> 0 bytes
 _build/codegen.cmi         |  Bin 5059 -> 0 bytes
 _build/codegen.cmo         |  Bin 17658 -> 0 bytes
 _build/codegen.cmx         |  Bin 1615 -> 0 bytes
 _build/codegen.ml          |  499 ----------------
 _build/codegen.ml.depends  |    1 -
 _build/codegen.o           |  Bin 87592 -> 0 bytes
 _build/javalite.cmi        |  Bin 795 -> 0 bytes
 _build/javalite.cmo        |  Bin 1956 -> 0 bytes
 _build/javalite.cmx        |  Bin 903 -> 0 bytes
 _build/javalite.ml         |   32 -
 _build/javalite.ml.depends |    1 -
 _build/javalite.native     |  Bin 1423656 -> 0 bytes
```

```
 _build/javalite.o          |  Bin 7960 -> 0 bytes
 _build/ocamlc.where        |    1 -
 _build/parser.cmi          |  Bin 1373 -> 0 bytes
 _build/parser.cmx          |  Bin 4700 -> 0 bytes
 _build/parser.ml           |  885 ------------------------
 _build/parser.ml.depends   |    1 -
 _build/parser.mli          |   45 --
 _build/parser.mli.depends  |    1 -
 _build/parser.mly          |  145 -----
 _build/parser.o            |  Bin 44024 -> 0 bytes
 _build/sast.cmi            |  Bin 2888 -> 0 bytes
 _build/sast.cmo            |  Bin 5667 -> 0 bytes
 _build/sast.cmx            |  Bin 620 -> 0 bytes
 _build/sast.ml             |  114 ----
 _build/sast.ml.depends     |    1 -
 _build/sast.o              |  Bin 25896 -> 0 bytes
 _build/scanner.cmi         |  Bin 1270 -> 0 bytes
 _build/scanner.cmo         |  Bin 21774 -> 0 bytes
 _build/scanner.cmx         |  Bin 20233 -> 0 bytes
 _build/scanner.ml          | 1528 ----------------------------------------------
 _build/scanner.ml.depends  |    1 -
 _build/scanner.mll         |   66 --
 _build/scanner.o           |  Bin 30608 -> 0 bytes
 _build/semant.cmi          |  Bin 7135 -> 0 bytes
 _build/semant.cmo          |  Bin 13720 -> 0 bytes
 _build/semant.cmx          |  Bin 2431 -> 0 bytes
 _build/semant.ml           |  397 ------------
 _build/semant.ml.depends   |    1 -
 _build/semant.o            |  Bin 68168 -> 0 bytes
 javalite.native            |    1 -
 50 files changed, 3972 deletions(-)

commit 131fe05d109210cad21617eb0425e047542d782e
Author: HongfeiChenCU <hc3222@columbia.edu>
Date:   Tue Apr 20 10:38:46 2021 -0700

    up to semant

 _build/_digests            |   32 +
 _build/_log                |   80 +++
 _build/ast.cmi             |  Bin 0 -> 3328 bytes
 _build/ast.cmo             |  Bin 0 -> 5857 bytes
 _build/ast.cmx             |  Bin 0 -> 687 bytes
 _build/ast.ml              |  139 ++++
 _build/ast.ml.depends      |    1 +
 _build/ast.o               |  Bin 0 -> 25760 bytes
 _build/codegen.cmi         |  Bin 0 -> 5059 bytes
 _build/codegen.cmo         |  Bin 0 -> 17658 bytes
 _build/codegen.cmx         |  Bin 0 -> 1615 bytes
 _build/codegen.ml          |  499 ++++++++++++++++
 _build/codegen.ml.depends  |    1 +
 _build/codegen.o           |  Bin 0 -> 87592 bytes
 _build/javalite.cmi        |  Bin 0 -> 795 bytes
 _build/javalite.cmo        |  Bin 0 -> 1956 bytes
 _build/javalite.cmx        |  Bin 0 -> 903 bytes
 _build/javalite.ml         |   32 +
 _build/javalite.ml.depends |    1 +
```

```
 _build/javalite.native     |  Bin 0 -> 1423656 bytes
 _build/javalite.o          |  Bin 0 -> 7960 bytes
 _build/ocamlc.where        |    1 +
 _build/parser.cmi          |  Bin 0 -> 1373 bytes
 _build/parser.cmx          |  Bin 0 -> 4700 bytes
 _build/parser.ml           |  885 ++++++++++++++++++++++++
 _build/parser.ml.depends   |    1 +
 _build/parser.mli          |   45 ++
 _build/parser.mli.depends  |    1 +
 _build/parser.mly          |  145 +++++
 _build/parser.o            |  Bin 0 -> 44024 bytes
 _build/sast.cmi            |  Bin 0 -> 2888 bytes
 _build/sast.cmo            |  Bin 0 -> 5667 bytes
 _build/sast.cmx            |  Bin 0 -> 620 bytes
 _build/sast.ml             |  114 ++++
 _build/sast.ml.depends     |    1 +
 _build/sast.o              |  Bin 0 -> 25896 bytes
 _build/scanner.cmi         |  Bin 0 -> 1270 bytes
 _build/scanner.cmo         |  Bin 0 -> 21774 bytes
 _build/scanner.cmx         |  Bin 0 -> 20233 bytes
 _build/scanner.ml          | 1528 ++++++++++++++++++++++++++++++++++++++++++++
 _build/scanner.ml.depends  |    1 +
 _build/scanner.mll         |   66 ++
 _build/scanner.o           |  Bin 0 -> 30608 bytes
 _build/semant.cmi          |  Bin 0 -> 7135 bytes
 _build/semant.cmo          |  Bin 0 -> 13720 bytes
 _build/semant.cmx          |  Bin 0 -> 2431 bytes
 _build/semant.ml           |  397 ++++++++++++
 _build/semant.ml.depends   |    1 +
 _build/semant.o            |  Bin 0 -> 68168 bytes
 ast.ml                     |   28 +-
 codegen.ml                 |    9 +-
 dev_tests/stest-var.jl     |    4 +
 javalite.native            |    1 +
 parser.mly                 |   24 +-
 sast.ml                    |    8 +-
 semant.ml                  |   41 +-
 tests/test-class4.jl       |    9 +-
 57 files changed, 4051 insertions(+), 44 deletions(-)

commit bb14d66b5acf2578f39f0a0abb8ef494dbb6c670
Merge: d58a198 b510aa8
Author: HongfeiChenCU <hc3222@columbia.edu>
Date:   Tue Apr 20 09:02:35 2021 -0700

    Merge branch 'main' of https://github.com/FranCao/javalite into main

commit d58a198ab64b6c4351e69f901d303bd7e1cce580
Author: HongfeiChenCU <hc3222@columbia.edu>
Date:   Tue Apr 20 09:02:18 2021 -0700

    string binop done

 codegen.ml                  | 30 +++++++++++++-------
 dev_tests/eqaulinc.c        |  8 ++++++
 semant.ml                   |  2 +-
 tests/test-class5.jl        | 62 ++++++++++++++++++++++++++++++++++++++++++
```

```
 tests/test-class5.out        |  7 +++++
 tests/test-string-binop2.jl  | 15 ++++++++++
 tests/test-string-binop2.out |  7 +++++
 7 files changed, 120 insertions(+), 11 deletions(-)

commit b510aa844210ccda421d3193b6e672f93abf3eab
Merge: e4eb6cf b70a0fd
Author: Frances Cao <frances.te.cao@gmail.com>
Date:   Mon Apr 19 21:41:57 2021 -0400

    resolve conflict

commit e4eb6cf9818d3e9013b7b1cef87210e106499715
Author: Frances Cao <frances.te.cao@gmail.com>
Date:   Mon Apr 19 21:40:49 2021 -0400

    start print array in codegen and semant, add test

 codegen.ml                 | 23 ++++++++++++++----------
 semant.ml                  | 12 ++++++++++-
 tests/test-array-func1.jl  |  8 ++++++++
 tests/test-array-func1.out |  1 +
 4 files changed, 33 insertions(+), 11 deletions(-)

commit b70a0fd96c3399d068524ddf459d0d521a5b716f
Author: 314pies <ian121363@gmail.com>
Date:   Mon Apr 19 17:41:30 2021 -0700

    Rename buildAndRun.sh and change its log file name

 buildAndRun.sh => BuildAndRun.sh | 4 ++--
 1 file changed, 2 insertions(+), 2 deletions(-)

commit 1db655421a19e18289a27d8298a375f35ea0fcc9
Author: 314pies <ian121363@gmail.com>
Date:   Mon Apr 19 17:36:36 2021 -0700

    Add BuildAndRun.sh for simply build, execute and show the execution result of
    ↪  the source code file. For example: You can run './BuildAndRun.sh
    ↪ SampleCode1.jl'  and it will show the execution result of SampleCode1.jl

 SampleCode1.jl |   7 +++
 buildAndRun.sh | 145 ++++++++++++++++++++++++++++++++++++++++++++++++++++++++
 2 files changed, 152 insertions(+)

commit aa492ec67a9c3979145819aa854b58fe61ccb4fe
Author: 314pies <ian121363@gmail.com>
Date:   Mon Apr 19 16:53:45 2021 -0700

    Add (optional) representative source language program - counting sort

 tests/test-counting-sort.jl  | 32 ++++++++++++++++++++++++++++++++
 tests/test-counting-sort.out | 15 +++++++++++++++
 2 files changed, 47 insertions(+)

commit badd72f7b972e1310cda7e28595b8e2237d0f38c
Author: 314pies <ian121363@gmail.com>
```

```
Date:   Mon Apr 19 16:14:35 2021 -0700

    Add performence measurement testing code

 tests/test-performence1.jl  | 14 +++++++++++++++
 tests/test-performence1.out |  1 +
 2 files changed, 15 insertions(+)

commit 8477bd4f62099db447646b6220e31abac8bef726
Author: 314pies <ian121363@gmail.com>
Date:   Mon Apr 19 15:54:27 2021 -0700

    Add representative source language program - bubble sort

 tests/test-bubble-sort.jl  | 27 +++++++++++++++++++++++++++
 tests/test-bubble-sort.out | 15 +++++++++++++++
 2 files changed, 42 insertions(+)

commit 70abb26e2460bdd3c512ac1f035f357a4276442a
Merge: 4e1867f a74452b
Author: HongfeiChenCU <hc3222@columbia.edu>
Date:   Sun Apr 18 23:16:34 2021 -0700

    Merge branch 'hc-class' into main

commit a74452bd6b44c9e745ee5ab3e4b735e109ed6711
Author: HongfeiChenCU <hc3222@columbia.edu>
Date:   Sun Apr 18 23:15:50 2021 -0700

    class done

 ast.ml                                          |  4 +
 codegen.ml                                      | 94 ++++++++++++++++++++----
 dev_tests/aatest-class.jl                       | 26 ++++++
 arrayinc.c => dev_tests/arrayinc.c              |  0
 ast-test-arrays.jl => dev_tests/ast-test-arrays.jl |  0
 dev_tests/classinc.c                            | 24 ++++++
 parser.mly                                      |  2 +-
 sast.ml                                         |  9 ++-
 semant.ml                                       | 55 ++++++++++------
 aatest-class.jl => tests/test-class1.jl         | 10 ++-
 tests/test-class1.out                           |  4 +
 tests/test-class2.jl                            | 24 ++++++
 tests/test-class2.out                           |  2 +
 tests/test-class3.jl                            | 40 +++++++++
 tests/test-class3.out                           |  2 +
 tests/test-class4.jl                            | 21 +++++
 tests/test-class4.out                           |  2 +
 17 files changed, 275 insertions(+), 44 deletions(-)

commit 2925e93ddfaff0d20879bff53b09b230019980ea
Author: HongfeiChenCU <hc3222@columbia.edu>
Date:   Sun Apr 18 12:45:42 2021 -0700

    class semant done

 Makefile              |  2 +-
```

```
 aatest-class.jl       | 14 +++++++++++
 ast.ml                | 19 +++++++++++---
 codegen.ml            |  2 +-
 parser.mly            | 30 +++++++++++-----------
 sast.ml               | 17 +++++++++++--
 scanner.mll           |  4 +--
 semant.ml             | 69 ++++++++++++++++++++++++++++++++++++++++++++++++---
 tests/fail-class1.err |  1 +
 tests/fail-class1.jl  | 14 +++++++++++
 tests/fail-class2.err |  1 +
 tests/fail-class2.jl  | 14 +++++++++++
 tests/test-array5.jl  | 19 ++++++++++++++
 tests/test-array5.out |  6 +++++
 14 files changed, 185 insertions(+), 27 deletions(-)

commit 4e1867f0038ef5bfbafa7bba9c52092d354a3ee5
Author: HongfeiChenCU <hc3222@columbia.edu>
Date:   Sun Apr 18 10:46:09 2021 -0700

    remove extra files commit by mistake

 _build/_digests            |   32 -
 _build/_log                |   52 --
 _build/ast.cmi             |  Bin 2836 -> 0 bytes
 _build/ast.cmo             |  Bin 4978 -> 0 bytes
 _build/ast.cmx             |  Bin 650 -> 0 bytes
 _build/ast.ml              |  120 ----
 _build/ast.ml.depends      |    1 -
 _build/ast.o               |  Bin 22296 -> 0 bytes
 _build/codegen.cmi         |  Bin 5089 -> 0 bytes
 _build/codegen.cmo         |  Bin 13584 -> 0 bytes
 _build/codegen.cmx         |  Bin 1611 -> 0 bytes
 _build/codegen.ml          |  410 -------------
 _build/codegen.ml.depends  |    1 -
 _build/codegen.o           |  Bin 67024 -> 0 bytes
 _build/javalite.cmi        |  Bin 767 -> 0 bytes
 _build/javalite.cmo        |  Bin 1928 -> 0 bytes
 _build/javalite.cmx        |  Bin 875 -> 0 bytes
 _build/javalite.ml         |   32 -
 _build/javalite.ml.depends |    1 -
 _build/javalite.native     |  Bin 1303776 -> 0 bytes
 _build/javalite.o          |  Bin 7960 -> 0 bytes
 _build/ocamlc.where        |    1 -
 _build/parser.cmi          |  Bin 1336 -> 0 bytes
 _build/parser.cmx          |  Bin 4371 -> 0 bytes
 _build/parser.ml           |  818 -------------------------
 _build/parser.ml.depends   |    1 -
 _build/parser.mli          |   43 --
 _build/parser.mli.depends  |    1 -
 _build/parser.mly          |  143 -----
 _build/parser.o            |  Bin 40640 -> 0 bytes
 _build/sast.cmi            |  Bin 2400 -> 0 bytes
 _build/sast.cmo            |  Bin 4622 -> 0 bytes
 _build/sast.cmx            |  Bin 581 -> 0 bytes
 _build/sast.ml             |   94 ---
 _build/sast.ml.depends     |    1 -
 _build/sast.o              |  Bin 21592 -> 0 bytes
```

```
 _build/scanner.cmi         |  Bin 1270 -> 0 bytes
 _build/scanner.cmo         |  Bin 20198 -> 0 bytes
 _build/scanner.cmx         |  Bin 18697 -> 0 bytes
 _build/scanner.ml          | 1421 ----------------------------------------------
 _build/scanner.ml.depends  |    1 -
 _build/scanner.mll         |   66 --
 _build/scanner.o           |  Bin 28968 -> 0 bytes
 _build/semant.cmi          |  Bin 4962 -> 0 bytes
 _build/semant.cmo          |  Bin 9976 -> 0 bytes
 _build/semant.cmx          |  Bin 1453 -> 0 bytes
 _build/semant.ml           |  288 ---------
 _build/semant.ml.depends   |    1 -
 _build/semant.o            |  Bin 50280 -> 0 bytes
 javalite.native            |    1 -
 testall.log                |  628 --------------------
 51 files changed, 4157 deletions(-)

commit 6876d88fd969477255fd712ee8c2e7b562178c1c
Author: HongfeiChenCU <hc3222@columbia.edu>
Date:   Sun Apr 18 10:42:35 2021 -0700

    array mutation test case added for string[]

 _build/_digests            |   32 +
 _build/_log                |   52 ++
 _build/ast.cmi             |  Bin 0 -> 2836 bytes
 _build/ast.cmo             |  Bin 0 -> 4978 bytes
 _build/ast.cmx             |  Bin 0 -> 650 bytes
 _build/ast.ml              |  120 ++++
 _build/ast.ml.depends      |    1 +
 _build/ast.o               |  Bin 0 -> 22296 bytes
 _build/codegen.cmi         |  Bin 0 -> 5089 bytes
 _build/codegen.cmo         |  Bin 0 -> 13584 bytes
 _build/codegen.cmx         |  Bin 0 -> 1611 bytes
 _build/codegen.ml          |  410 ++++++++++++++
 _build/codegen.ml.depends  |    1 +
 _build/codegen.o           |  Bin 0 -> 67024 bytes
 _build/javalite.cmi        |  Bin 0 -> 767 bytes
 _build/javalite.cmo        |  Bin 0 -> 1928 bytes
 _build/javalite.cmx        |  Bin 0 -> 875 bytes
 _build/javalite.ml         |   32 +
 _build/javalite.ml.depends |    1 +
 _build/javalite.native     |  Bin 0 -> 1303776 bytes
 _build/javalite.o          |  Bin 0 -> 7960 bytes
 _build/ocamlc.where        |    1 +
 _build/parser.cmi          |  Bin 0 -> 1336 bytes
 _build/parser.cmx          |  Bin 0 -> 4371 bytes
 _build/parser.ml           |  818 ++++++++++++++++++++++++++
 _build/parser.ml.depends   |    1 +
 _build/parser.mli          |   43 ++
 _build/parser.mli.depends  |    1 +
 _build/parser.mly          |  143 +++++
 _build/parser.o            |  Bin 0 -> 40640 bytes
 _build/sast.cmi            |  Bin 0 -> 2400 bytes
 _build/sast.cmo            |  Bin 0 -> 4622 bytes
 _build/sast.cmx            |  Bin 0 -> 581 bytes
 _build/sast.ml             |   94 +++
```

```
 _build/sast.ml.depends     |    1 +
 _build/sast.o              |  Bin 0 -> 21592 bytes
 _build/scanner.cmi         |  Bin 0 -> 1270 bytes
 _build/scanner.cmo         |  Bin 0 -> 20198 bytes
 _build/scanner.cmx         |  Bin 0 -> 18697 bytes
 _build/scanner.ml          | 1421 ++++++++++++++++++++++++++++++++++++++++++++++++
 _build/scanner.ml.depends  |    1 +
 _build/scanner.mll         |   66 ++
 _build/scanner.o           |  Bin 0 -> 28968 bytes
 _build/semant.cmi          |  Bin 0 -> 4962 bytes
 _build/semant.cmo          |  Bin 0 -> 9976 bytes
 _build/semant.cmx          |  Bin 0 -> 1453 bytes
 _build/semant.ml           |  288 +++++++++
 _build/semant.ml.depends   |    1 +
 _build/semant.o            |  Bin 0 -> 50280 bytes
 javalite.native            |    1 +
 testall.log                |  628 ++++++++++++++++++++
 tests/test-array5.jl       |   19 +
 tests/test-array5.out      |    6 +
 53 files changed, 4182 insertions(+)

commit fb6832d8fa3d91cd82078327d08b1281ed5c0d42
Author: HongfeiChenCU <hc3222@columbia.edu>
Date:   Sun Apr 18 10:31:46 2021 -0700

    string mutation done

 aatest-class.jl       |  0
 ast-test-arrays.jl    |  3 +++
 ast.ml                |  2 ++
 codegen.ml            | 12 ++++++++++-
 parser.mly            | 18 ++++++++++++++++++
 sast.ml               |  2 ++
 scanner.mll           |  2 ++
 semant.ml             | 14 ++++++++++++++-
 tests/fail-array4.err |  1 +
 tests/fail-array4.jl  |  6 ++++++
 tests/test-array4.jl  | 19 +++++++++++++++++++
 tests/test-array4.out |  6 ++++++
 12 files changed, 83 insertions(+), 2 deletions(-)

commit de71fe27cd9e336f2a63be684b37f75c1cd6bc0f
Author: HongfeiChenCU <hc3222@columbia.edu>
Date:   Fri Apr 16 16:01:04 2021 -0700

    string indexof fixed

 stringfuncs.c               |   6 +++---
 stringfuncs.o               | Bin 3272 -> 3272 bytes
 tests/test-afunc-string5.jl |   6 +++++-
 tests/test-afunc-string5.out |   3 ++-
 4 files changed, 10 insertions(+), 5 deletions(-)

commit ce589b39600c4b80727cd15c4d6f199aca600a0c
Merge: 5fe0ddf 784a98f
Author: Frances Cao <frances.te.cao@gmail.com>
Date:   Thu Apr 15 20:39:49 2021 -0400
```

```
    Merge branch 'main' of github.com:FranCao/javalite into main

commit 5fe0ddfecaebad72d85aa8d25bc722490d9ef656
Author: Frances Cao <frances.te.cao@gmail.com>
Date:   Thu Apr 15 20:39:42 2021 -0400

    combine funcs, add concat, len, and indexof funcs for string

 codegen.ml                                        |  21 +++++++++++++
 semant.ml                                         |  28 ++++++++++++++++++
 stringfuncs.c                                     |  35 ++++++++++++++++++++++
 stringfuncs.o                                     | Bin 2528 -> 3272 bytes
 tests/test-afunc-string3.jl                       |   5 +++
 tests/test-afunc-string3.out                      |   3 +-
 tests/test-afunc-string5.jl                       |  25 ++++++++++++++++
 tests/test-afunc-string5.out                      |   4 +++
 ...est-astring-binop1.jl => test-astring-binop.jl} |   0
 ...t-astring-binop1.out => test-astring-binop.out} |   0
 tests/test-astring-binop2.jl                      |   8 -----
 tests/test-astring-binop2.out                     |   1 -
 12 files changed, 120 insertions(+), 10 deletions(-)

commit 784a98f4e4ebfeee935c130f55bb2164d4e25a10
Author: HongfeiChenCU <hc3222@columbia.edu>
Date:   Thu Apr 15 15:49:17 2021 -0700

    print for arrayaccess added

 codegen.ml            |  2 +-
 tests/test-array1.jl  |  4 +---
 tests/test-array3.jl  | 11 ++++++++++
 tests/test-array3.out |  2 ++
 4 files changed, 15 insertions(+), 4 deletions(-)

commit 1922ed0e773974bc34abdfef7eb9d830b50abc4c
Merge: 955689f bc6d10f
Author: Frances Cao <frances.te.cao@gmail.com>
Date:   Thu Apr 15 17:27:55 2021 -0400

    Merge branch 'main' of github.com:FranCao/javalite into main

commit 955689f14a95db3a45c0ac7460e1d9e8de445bb3
Author: Frances Cao <frances.te.cao@gmail.com>
Date:   Thu Apr 15 17:27:51 2021 -0400

    concatenate built in functions to one file

 Makefile                                     |  20 ++-------
 _tags                                        |  13 +-----
 codegen.ml                                   |   5 +++
 reversestring.c => funcs/reversestring.c     |   0
 stringlower.c => funcs/stringlower.c         |   0
 stringsubstring.c => funcs/stringsubstring.c |   0
 stringupper.c => funcs/stringupper.c         |   0
 reversestring.o                              | Bin 1592 -> 0 bytes
 sast.ml                                      |   4 ++
```

```
 semant.ml                                    |   2 +-
 stringfuncs.c                                |  63 ++++++++++++++++++++++++++
 stringfuncs.o                                | Bin 0 -> 2528 bytes
 stringlower.o                                | Bin 1648 -> 0 bytes
 stringsubstring.o                            | Bin 1608 -> 0 bytes
 stringupper.o                                | Bin 1648 -> 0 bytes
 testall.sh                                   |  27 ++----------
 16 files changed, 83 insertions(+), 51 deletions(-)

commit bc6d10f7d6aa1ce8601aea6d34e6653f0a7da0d0
Author: HongfeiChenCU <hc3222@columbia.edu>
Date:   Thu Apr 15 12:24:09 2021 -0700

    small array fix

 codegen.ml | 4 +---
 1 file changed, 1 insertion(+), 3 deletions(-)

commit 0344481221c28b92937dfd7fe3fbec7cc30efd38
Author: HongfeiChenCU <hc3222@columbia.edu>
Date:   Wed Apr 14 16:41:42 2021 -0700

    array lit and access done

 codegen.ml            | 33 ++++++++++++++++++++++++++--------------
 tests/test-array2.jl  | 13 +++++++++++++
 tests/test-array2.out |  5 +++++
 3 files changed, 37 insertions(+), 14 deletions(-)

commit cf08e0c2c31f15ec1d5172830bc781a27556b120
Author: HongfeiChenCU <hc3222@columbia.edu>
Date:   Wed Apr 14 15:48:02 2021 -0700

    array basic done

 arrayinc.c              |  4 ++++
 ast.ml                  |  2 +-
 codegen.ml              | 40 ++++++++++++++++++++++++++++++++++------
 parser.mly              |  2 +-
 sast.ml                 |  2 +-
 tests/test-aglobal4.jl  |  7 +++++++
 tests/test-aglobal4.out |  2 ++
 tests/test-array1.jl    | 15 +++++++++++++++
 tests/test-array1.out   |  4 ++++
 9 files changed, 69 insertions(+), 9 deletions(-)

commit 528a48b65ca698caecab1e7db9d46c6e7f9d4dd9
Author: Frances Cao <frances.te.cao@gmail.com>
Date:   Sun Apr 11 19:30:00 2021 -0400

    add string binop equals

 tests/test-afunc-string5.jl   |  6 ------
 tests/test-afunc-string5.out  |  1 -
 tests/test-astring-binop.jl   | 14 --------------
 tests/test-astring-binop.out  |  3 ---
 tests/test-astring-binop1.jl  |  6 ++++++
```

```
 tests/test-astring-binop1.out |  1 +
 tests/test-astring-binop2.jl  |  8 ++++++++
 tests/test-astring-binop2.out |  1 +
 8 files changed, 16 insertions(+), 24 deletions(-)

commit 09fad9a4c8942809ffb2bdbd3a81e88aab0f66b2
Merge: 6bdb993 4565f0e
Author: HongfeiChenCU <hc3222@columbia.edu>
Date:   Sun Apr 11 16:12:06 2021 -0700

    merge string binop

commit 6bdb99309c0c8c24380aadb570a1e7a1cd1bfd22
Author: HongfeiChenCU <hc3222@columbia.edu>
Date:   Sun Apr 11 16:10:29 2021 -0700

    string addition added

 codegen.ml                    | 26 ++++++++++++++++++++++----
 sast.ml                       |  6 +++++-
 semant.ml                     | 10 ++++++++++
 tests/test-afunc-string5.jl   |  6 ++++++
 tests/test-afunc-string5.out |  1 +
 5 files changed, 44 insertions(+), 5 deletions(-)

commit 4565f0e4d3fb26a86a42b2812c0b595bb0b3688f
Author: Frances Cao <frances.te.cao@gmail.com>
Date:   Sun Apr 11 00:59:32 2021 -0400

    minor print changes, add test for string binop

 codegen.ml                    | 16 ++++++++++++----
 tests/test-astring-binop.jl   | 14 ++++++++++++++
 tests/test-astring-binop.out |  3 +++
 3 files changed, 29 insertions(+), 4 deletions(-)

commit f2910da50b94dd1ed7f3ba7ddc20614f50f98cff
Merge: 61edc04 28dc68c
Author: HongfeiChenCU <hc3222@columbia.edu>
Date:   Fri Apr 9 23:25:38 2021 -0700

    merged array changes

commit 61edc04af63e3a6eb75da6e904991bb4293b6258
Author: HongfeiChenCU <hc3222@columbia.edu>
Date:   Sat Apr 10 06:18:32 2021 +0000

    print done

 Makefile                     |  2 +-
 ast.ml                       |  4 +--
 codegen.ml                   | 72 ++++++++++++++++----------------------------------
 semant.ml                    |  4 +--
 tests/test-afunc-string2.jl |  2 +-
 tests/test-afunc-string4.jl |  2 +-
 tests/test-helloworld.jl    |  2 +-
 7 files changed, 33 insertions(+), 55 deletions(-)
```

```
commit ce81a3a827bca3107bab07ad36ae1df2bd7a6e6c
Author: HongfeiChenCU <hc3222@columbia.edu>
Date:   Fri Apr 9 21:58:41 2021 -0700

    print working except string and double

 Makefile                    |   7 ++-
 ast.ml                      |   3 +-
 codegen.ml                  | 103 ++++++++++++++++++++++++++++++++-------------
 semant.ml                   |  12 +++---
 tests/fail-printb.err       |   1 -
 tests/fail-printb.jl        |   2 -
 tests/test-afunc-string1.jl |   6 +--
 tests/test-afunc-string2.jl |   2 +-
 tests/test-afunc-string3.jl |   4 +-
 tests/test-afunc-string4.jl |   8 ++--
 tests/test-double1.jl       |   2 +-
 tests/test-double2.jl       |   2 +-
 tests/test-double3.jl       |  24 +++++------
 tests/test-ops1.jl          |  28 ++++++------
 tests/test-ops2.jl          |  24 +++++------
 tests/test-printstring.jl   |   2 +-
 tests/test-var3.jl          |   2 +-
 17 files changed, 138 insertions(+), 94 deletions(-)

commit aca4bb28c4e8b2799fc30df25886d7fc0439ecd6
Author: Frances Cao <frances.te.cao@gmail.com>
Date:   Fri Apr 9 00:34:44 2021 -0400

    complete substring built-in function

 codegen.ml                  |  34 +++++++++++++++++++++++++++++++--
 semant.ml                   |  12 ++++++++++--
 stringsubstring.c           |   6 +++---
 stringsubstring.o           | Bin 1600 -> 1608 bytes
 test-afunc-string4.ll       |   0
 tests/test-afunc-string4.jl |   4 ++--
 tests/test-afunc-string4.out |   3 +--
 7 files changed, 48 insertions(+), 11 deletions(-)

commit 01a20e573971a33abf28cb3cd075332415351c6e
Author: Frances Cao <frances.te.cao@gmail.com>
Date:   Thu Apr 8 20:54:38 2021 -0400

    substring edits

 codegen.ml                  | 6 +++---
 test-afunc-string4.ll       | 0
 tests/test-afunc-string4.jl | 4 ++--
 tests/test-afunc-string4.out | 2 +-
 4 files changed, 6 insertions(+), 6 deletions(-)

commit b578d1e86ccab755b6e5e5e9c2dbbb22e475f769
Author: Frances Cao <frances.te.cao@gmail.com>
Date:   Thu Apr 8 20:13:05 2021 -0400

```

```
    add substring func

 Makefile                     |   6 +++++-
 _tags                        |   5 ++++-
 codegen.ml                   |   7 +++++++
 semant.ml                    |   3 ++-
 stringsubstring.c            |  31 ++++++++++++++++++++++++++++++++
 stringsubstring.o            | Bin 0 -> 1600 bytes
 testall.sh                   |   9 ++++++++-
 tests/test-afunc-string4.jl  |  19 +++++++++++++++++++
 tests/test-afunc-string4.out |   6 ++++++
 9 files changed, 82 insertions(+), 4 deletions(-)

commit 28dc68c56f7b47b126728903c526a48a912b94bd
Author: HongfeiChenCU <hc3222@columbia.edu>
Date:   Thu Apr 8 01:47:08 2021 -0700

    allow 2d arrays

 ast-test-arrays.jl | 17 ++++++++++------
 ast.ml             | 10 +++++------
 parser.mly         |  8 +++++---
 scanner.mll        |  5 +++-
 semant.ml          | 18 ++++++++++++---------
 5 files changed, 22 insertions(+), 36 deletions(-)

commit f7c322d464d16a33ea9219e854f2b56e7ebf6274
Merge: b7255e5 b8cdd73
Author: Frances Cao <frances.te.cao@gmail.com>
Date:   Wed Apr 7 21:34:48 2021 -0400

    Merge branch 'main' of github.com:FranCao/javalite into main

commit b7255e5d76082167e53e04552d4c02fae15c2280
Author: Frances Cao <frances.te.cao@gmail.com>
Date:   Wed Apr 7 21:34:36 2021 -0400

    cleanup printbig

 Makefile                |  15 +++++------
 _tags                   |   3 --
 codegen.ml              |   7 ----
 printbig.c              |  75 -----------------------------------------
 printbig.o              | Bin 2080 -> 0 bytes
 semant.ml               |   3 +-
 testall.sh              |   9 +++++-
 tests/fail-printbig.err |   1 -
 tests/fail-printbig.jl  |   2 --
 tests/test-printbig.jl  |  25 ----------
 tests/test-printbig.out |  88 -------------------------------------------------
 11 files changed, 7 insertions(+), 221 deletions(-)

commit b8cdd73554829c81562b578abdd7d018d26e0792
Author: HongfeiChenCU <hc3222@columbia.edu>
Date:   Wed Apr 7 18:26:32 2021 -0700

    class part deleted
```

```
 ast-test-class.jl | 26 -------------------------
 ast.ml            | 36 ++++---------------------------------
 parser.mly        | 38 +++-----------------------------------
 scanner.mll       |  4 ----
 semant.ml         |  9 +++++----
 5 files changed, 12 insertions(+), 101 deletions(-)

commit 4f81c56468f7778f89a585c5dd58e856bedd8921
Merge: e0789fc cfcc69f
Author: Frances Cao <frances.te.cao@gmail.com>
Date:   Wed Apr 7 21:00:03 2021 -0400

    merge

commit e0789fc12a3924d58bd32d574ab9d6bad0587329
Author: Frances Cao <frances.te.cao@gmail.com>
Date:   Wed Apr 7 20:56:45 2021 -0400

    edit tests for print functions

 _build/_digests              |   32 -
 _build/_log                  |   88 ---
 _build/ast.cmi               |  Bin 3461 -> 0 bytes
 _build/ast.cmo               |  Bin 6272 -> 0 bytes
 _build/ast.cmx               |  Bin 690 -> 0 bytes
 _build/ast.ml                |  149 ----
 _build/ast.ml.depends        |    1 -
 _build/ast.o                 |  Bin 26904 -> 0 bytes
 _build/codegen.cmi           |  Bin 5089 -> 0 bytes
 _build/codegen.cmo           |  Bin 9365 -> 0 bytes
 _build/codegen.cmx           |  Bin 1589 -> 0 bytes
 _build/codegen.ml            |  275 -------
 _build/codegen.ml.depends    |    1 -
 _build/codegen.o             |  Bin 47984 -> 0 bytes
 _build/javalite.cmi          |  Bin 767 -> 0 bytes
 _build/javalite.cmo          |  Bin 1928 -> 0 bytes
 _build/javalite.cmx          |  Bin 875 -> 0 bytes
 _build/javalite.ml           |   32 -
 _build/javalite.ml.depends   |    1 -
 _build/javalite.native       |  Bin 1296344 -> 0 bytes
 _build/javalite.o            |  Bin 7960 -> 0 bytes
 _build/ocamlc.where          |    1 -
 _build/parser.cmi            |  Bin 1486 -> 0 bytes
 _build/parser.cmx            |  Bin 5086 -> 0 bytes
 _build/parser.ml             |  982 ----------------------
 _build/parser.ml.depends     |    1 -
 _build/parser.mli            |   50 --
 _build/parser.mli.depends    |    1 -
 _build/parser.mly            |  159 ----
 _build/parser.o              |  Bin 48432 -> 0 bytes
 _build/sast.cmi              |  Bin 2105 -> 0 bytes
 _build/sast.cmo              |  Bin 4015 -> 0 bytes
 _build/sast.cmx              |  Bin 505 -> 0 bytes
 _build/sast.ml               |   84 --
 _build/sast.ml.depends       |    1 -
 _build/sast.o                |  Bin 18672 -> 0 bytes
```

```
 _build/scanner.cmi           |  Bin 1270 -> 0 bytes
 _build/scanner.cmo           |  Bin 25894 -> 0 bytes
 _build/scanner.cmx           |  Bin 24253 -> 0 bytes
 _build/scanner.ml            | 1803 ----------------------------------------
 _build/scanner.ml.depends    |    1 -
 _build/scanner.mll           |   71 --
 _build/scanner.o             |  Bin 34872 -> 0 bytes
 _build/semant.cmi            |  Bin 4974 -> 0 bytes
 _build/semant.cmo            |  Bin 7577 -> 0 bytes
 _build/semant.cmx            |  Bin 1456 -> 0 bytes
 _build/semant.ml             |  201 -----
 _build/semant.ml.depends     |    1 -
 _build/semant.o              |  Bin 38616 -> 0 bytes
 javalite.native              |    1 -
 testall.log                  |  554 -------------
 tests/test-afunc-string1.jl  |    2 +
 tests/test-afunc-string1.out |    3 +-
 53 files changed, 4 insertions(+), 4491 deletions(-)

commit cfcc69fd883a91df10acbf142586baa96e0d6e20
Merge: 677189f 1a291fe
Author: HongfeiChenCU <hc3222@columbia.edu>
Date:   Wed Apr 7 17:53:23 2021 -0700

    Merge branch 'main' of https://github.com/FranCao/javalite into main

commit 677189fa155ce6f96df9ef7d11e44456b8a2995a
Merge: e498e6f 805c414
Author: HongfeiChenCU <hc3222@columbia.edu>
Date:   Wed Apr 7 17:27:59 2021 -0700

    merge arr

commit e498e6f6ad5ee5497f1bf32e1cb61c221cd87eb6
Author: HongfeiChenCU <hc3222@columbia.edu>
Date:   Mon Apr 5 21:38:56 2021 -0700

    array access parser updated

 ast-test-arrays.jl | 2 +-
 ast.ml             | 6 +++---
 codegen.ml         | 2 +-
 parser.mly         | 2 +-
 sast.ml            | 6 +++---
 semant.ml          | 8 +++++++--
 6 files changed, 15 insertions(+), 11 deletions(-)

commit a2ecfec99cc19fa88873d5493507673bf070fe50
Author: HongfeiChenCU <hc3222@columbia.edu>
Date:   Mon Apr 5 16:41:29 2021 -0700

    array semant done

 ast-test-arrays.jl    | 12 +++++++-----
 ast.ml                | 18 +++++++++---------
 codegen.ml            |  1 +
 parser.mly            |  6 +++---
```

```
 sast.ml               |  4 ++--
 semant.ml             | 36 ++++++++++++++++++++++++++++++++++++
 tests/fail-array1.err |  1 +
 tests/fail-array1.jl  |  5 +++++
 tests/fail-array2.err |  1 +
 tests/fail-array2.jl  |  5 +++++
 tests/fail-array3.err |  1 +
 tests/fail-array3.jl  | 10 ++++++++++
 12 files changed, 81 insertions(+), 19 deletions(-)

commit 9a656fb885987dd5e45d238039f1c3a243090bd4
Author: Frances Cao <frances.te.cao@gmail.com>
Date:   Sun Apr 4 18:30:39 2021 -0400

    add upper and lower functions for string

 Makefile                     |   10 +-
 _build/_digests              |   32 +
 _build/_log                  |   88 ++-
 _build/ast.cmi               |  Bin 0 -> 3461 bytes
 _build/ast.cmo               |  Bin 0 -> 6272 bytes
 _build/ast.cmx               |  Bin 0 -> 690 bytes
 _build/ast.ml                |  149 ++++
 _build/ast.ml.depends        |    1 +
 _build/ast.o                 |  Bin 0 -> 26904 bytes
 _build/codegen.cmi           |  Bin 0 -> 5089 bytes
 _build/codegen.cmo           |  Bin 0 -> 9365 bytes
 _build/codegen.cmx           |  Bin 0 -> 1589 bytes
 _build/codegen.ml            |  275 +++++++
 _build/codegen.ml.depends    |    1 +
 _build/codegen.o             |  Bin 0 -> 47984 bytes
 _build/javalite.cmi          |  Bin 0 -> 767 bytes
 _build/javalite.cmo          |  Bin 0 -> 1928 bytes
 _build/javalite.cmx          |  Bin 0 -> 875 bytes
 _build/javalite.ml           |   32 +
 _build/javalite.ml.depends   |    1 +
 _build/javalite.native       |  Bin 0 -> 1296344 bytes
 _build/javalite.o            |  Bin 0 -> 7960 bytes
 _build/ocamlc.where          |    1 +
 _build/parser.cmi            |  Bin 0 -> 1486 bytes
 _build/parser.cmx            |  Bin 0 -> 5086 bytes
 _build/parser.ml             |  982 ++++++++++++++++++++++++
 _build/parser.ml.depends     |    1 +
 _build/parser.mli            |   50 ++
 _build/parser.mli.depends    |    1 +
 _build/parser.mly            |  159 ++++
 _build/parser.o              |  Bin 0 -> 48432 bytes
 _build/sanitize.sh           |    8 -
 _build/sast.cmi              |  Bin 0 -> 2105 bytes
 _build/sast.cmo              |  Bin 0 -> 4015 bytes
 _build/sast.cmx              |  Bin 0 -> 505 bytes
 _build/sast.ml               |   84 ++
 _build/sast.ml.depends       |    1 +
 _build/sast.o                |  Bin 0 -> 18672 bytes
 _build/scanner.cmi           |  Bin 0 -> 1270 bytes
 _build/scanner.cmo           |  Bin 0 -> 25894 bytes
 _build/scanner.cmx           |  Bin 0 -> 24253 bytes
```

```
 _build/scanner.ml            | 1803 +++++++++++++++++++++++++++++++++++++++++++
 _build/scanner.ml.depends    |    1 +
 _build/scanner.mll           |   71 ++
 _build/scanner.o             |  Bin 0 -> 34872 bytes
 _build/semant.cmi            |  Bin 0 -> 4974 bytes
 _build/semant.cmo            |  Bin 0 -> 7577 bytes
 _build/semant.cmx            |  Bin 0 -> 1456 bytes
 _build/semant.ml             |  201 +++++
 _build/semant.ml.depends     |    1 +
 _build/semant.o              |  Bin 0 -> 38616 bytes
 _tags                        |    8 +
 codegen.ml                   |   20 +-
 javalite.native              |    1 +
 reversestring.c              |    4 +-
 reversestring.o              |  Bin 1600 -> 1592 bytes
 semant.ml                    |    4 +-
 stringlower.c                |   28 +
 stringlower.o                |  Bin 0 -> 1648 bytes
 stringupper.c                |   28 +
 stringupper.o                |  Bin 0 -> 1648 bytes
 testall.log                  |  554 ++++++++++++++
 testall.sh                   |   16 +-
 tests/test-afunc-string2.jl  |    9 +
 tests/test-afunc-string2.out |    2 +
 tests/test-afunc-string3.jl  |    9 +
 tests/test-afunc-string3.out |    2 +
 67 files changed, 4621 insertions(+), 17 deletions(-)

commit 603a2ee85c25ddfda666774aa73198cb84f51a4e
Author: Frances Cao <frances.te.cao@gmail.com>
Date:   Sun Apr 4 16:59:24 2021 -0400

    add reverse

 Makefile                     |   6 +++++-
 README                       |   2 +-
 _build/_log                  |   2 ++
 _build/sanitize.sh           |   8 ++++++++
 codegen.ml                   |   9 ++++++++-
 reversestring.c              |  27 +++++++++++++++++++++++++++
 reversestring.o              | Bin 0 -> 1600 bytes
 semant.ml                    |  13 ++++++++++---
 testall.sh                   |   9 ++++++++-
 tests/test-afunc-string1.jl  |   9 +++++++++
 tests/test-afunc-string1.out |   2 ++
 11 files changed, 80 insertions(+), 7 deletions(-)

commit 1a291fe310ba120ab6e1fcc8e2bf9b3634431c47
Merge: 543d4c3 805c414
Author: Mateo Maturana <48253525+maturanamateo@users.noreply.github.com>
Date:   Sun Apr 4 14:09:19 2021 -0400

    Merge pull request #1 from FranCao/arrays

    Arrays

commit 805c41456bc7f5b812eaacdc6676ac889b56277f
```

```
Author: Mateo Maturana <mateomaturana@yahoo.com>
Date:   Sun Apr 4 13:24:32 2021 -0400

    minor changes

 ast.ml    | 2 +-
 semant.ml | 2 ++
 2 files changed, 3 insertions(+), 1 deletion(-)

commit ab788cc135fdbee3dd5e5e9fc045aa50f964265d
Author: Mateo Maturana <mateomaturana@yahoo.com>
Date:   Sat Mar 27 17:55:21 2021 -0400

    finished sast

 sast.ml | 5 +++++
 1 file changed, 5 insertions(+)

commit 622ab5357df0a5c7c3b7c6a11879c97afaf5d7d4
Author: Mateo Maturana <mateomaturana@yahoo.com>
Date:   Sat Mar 27 13:41:35 2021 -0400

    improvements to pretty printing, mandating int for arr access, not expr

 ast-test-arrays.jl | 2 +-
 ast.ml             | 6 +++---
 parser.mly         | 2 +-
 3 files changed, 5 insertions(+), 5 deletions(-)

commit c8d647564132f3141fe93f7eb995668b81591daf
Author: Mateo Maturana <mateomaturana@yahoo.com>
Date:   Sat Mar 27 13:24:45 2021 -0400

    arrays complete in ast, parser, scanner

 ast-test-arrays.jl |  4 +++-
 ast.ml             | 10 +++++++---
 parser.mly         | 12 ++++++------
 scanner.mll        |  4 ++++
 4 files changed, 20 insertions(+), 10 deletions(-)

commit 093fe067fc1e1681f96f30f9c6a60048f9fddb9d
Author: Mateo Maturana <mateomaturana@yahoo.com>
Date:   Fri Mar 26 06:29:23 2021 -0400

    arrays partially implemented

 ast-test-arrays.jl |  6 ++++++
 ast.ml             | 10 +++++-----
 parser.mly         |  3 ++-
 3 files changed, 13 insertions(+), 6 deletions(-)

commit f223498628771f72baeac1f920b47a3df37128a5
Author: Mateo Maturana <mateomaturana@yahoo.com>
Date:   Fri Mar 26 06:03:05 2021 -0400

    ast, parser, scnaner for array prelim
```

```
 ast.ml      | 10 +++++++---
 parser.mly  |  8 +++++++-
 scanner.mll |  2 ++
 3 files changed, 16 insertions(+), 4 deletions(-)

commit 543d4c3e4abbd968aebf46b82990e2604b9a5133
Author: HongfeiChenCU <hc3222@columbia.edu>
Date:   Wed Mar 24 21:10:46 2021 -0700

    CLASS parser, sanner, ast finished

 ast-test-class.jl   | 26 ++++++++++++++++++++++++
 ast.ml              | 38 ++++++++++++++++++++++++++++++++++++---
 codegen.ml          | 11 +++++++----
 parser.mly          | 40 ++++++++++++++++++++++++++++++++++++---
 scanner.mll         | 14 ++++++++------
 semant.ml           |  2 +-
 tests/test-var3.jl  | 10 ++++++++++
 tests/test-var3.out |  2 ++
 8 files changed, 126 insertions(+), 17 deletions(-)

commit e82ab35a20f1ea9f2f53bfcfb89fc09fda30e8c2
Author: Frances Cao <frances.te.cao@gmail.com>
Date:   Mon Mar 22 21:43:40 2021 -0400

    add single line comments, edit test cases, replace float with double

 ast.ml                                    | 16 ++++++------
 codegen.ml                                | 16 ++++++------
 parser.mly                                | 41 ++++++++++++++++++--------------
 sast.ml                                   | 12 ++++++----
 scanner.mll                               | 24 ++++++++++-------
 semant.ml                                 | 14 +++++++-----
 tests/fail-double1.err                    |  1 +
 tests/fail-double1.jl                     |  5 ++++
 tests/fail-double2.err                    |  1 +
 tests/fail-double2.jl                     |  5 ++++
 tests/fail-expr3.err                      |  2 +-
 tests/fail-expr3.jl                       |  8 +++++---
 tests/fail-float1.err                     |  1 -
 tests/fail-float1.jl                      |  5 ----
 tests/fail-float2.err                     |  1 -
 tests/fail-float2.jl                      |  5 ----
 tests/{test-float1.jl => test-double1.jl} |  2 +-
 tests/{test-float1.out => test-double1.out} |  0
 tests/{test-float2.jl => test-double2.jl} |  6 ++++--
 tests/{test-float2.out => test-double2.out} |  0
 tests/{test-float3.jl => test-double3.jl} | 10 +++++----
 tests/{test-float3.out => test-double3.out} |  0
 tests/test-printstring.jl                 |  6 +++++
 tests/test-printstring.out                |  1 +
 24 files changed, 97 insertions(+), 85 deletions(-)

commit fde4100ebc3b2cd081c2bd415ebacb681915744d
Author: HongfeiChenCU <hc3222@columbia.edu>
Date:   Sun Mar 21 20:47:57 2021 -0700
```

```
2112
2113     renamed all files for javalite
2114
2115  Makefile                                          | 20 +++++++++---------
2116  codegen.ml                                        |  2 +-
2117  microc.ml => javalite.ml                          |  6 +++---
2118  microcparse.mly => parser.mly                     |  4 ++--
2119  scanner.mll                                       |  2 +-
2120  testall.sh                                        | 26 ++++++++++++------------
2121  tests/{fail-assign1.mc => fail-assign1.jl}        |  0
2122  tests/{fail-assign2.mc => fail-assign2.jl}        |  0
2123  tests/{fail-assign3.mc => fail-assign3.jl}        |  0
2124  tests/{fail-dead1.mc => fail-dead1.jl}            |  0
2125  tests/{fail-dead2.mc => fail-dead2.jl}            |  0
2126  tests/{fail-expr1.mc => fail-expr1.jl}            |  0
2127  tests/{fail-expr2.mc => fail-expr2.jl}            |  0
2128  tests/{fail-expr3.mc => fail-expr3.jl}            |  0
2129  tests/{fail-float1.mc => fail-float1.jl}          |  0
2130  tests/{fail-float2.mc => fail-float2.jl}          |  0
2131  tests/{fail-for1.mc => fail-for1.jl}              |  0
2132  tests/{fail-for2.mc => fail-for2.jl}              |  0
2133  tests/{fail-for3.mc => fail-for3.jl}              |  0
2134  tests/{fail-for4.mc => fail-for4.jl}              |  0
2135  tests/{fail-for5.mc => fail-for5.jl}              |  0
2136  tests/{fail-func1.mc => fail-func1.jl}            |  0
2137  tests/{fail-func2.mc => fail-func2.jl}            |  0
2138  tests/{fail-func3.mc => fail-func3.jl}            |  0
2139  tests/{fail-func4.mc => fail-func4.jl}            |  0
2140  tests/{fail-func5.mc => fail-func5.jl}            |  0
2141  tests/{fail-func6.mc => fail-func6.jl}            |  0
2142  tests/{fail-func7.mc => fail-func7.jl}            |  0
2143  tests/{fail-func8.mc => fail-func8.jl}            |  0
2144  tests/{fail-func9.mc => fail-func9.jl}            |  0
2145  tests/{fail-global1.mc => fail-global1.jl}        |  0
2146  tests/{fail-global2.mc => fail-global2.jl}        |  0
2147  tests/{fail-if1.mc => fail-if1.jl}                |  0
2148  tests/{fail-if2.mc => fail-if2.jl}                |  0
2149  tests/{fail-if3.mc => fail-if3.jl}                |  0
2150  tests/{fail-nomain.mc => fail-nomain.jl}          |  0
2151  tests/{fail-print.mc => fail-print.jl}            |  0
2152  tests/{fail-printb.mc => fail-printb.jl}          |  0
2153  tests/{fail-printbig.mc => fail-printbig.jl}      |  0
2154  tests/{fail-return1.mc => fail-return1.jl}        |  0
2155  tests/{fail-return2.mc => fail-return2.jl}        |  0
2156  tests/{fail-while1.mc => fail-while1.jl}          |  0
2157  tests/{fail-while2.mc => fail-while2.jl}          |  0
2158  tests/{test-add1.mc => test-add1.jl}              |  0
2159  tests/{test-arith1.mc => test-arith1.jl}          |  0
2160  tests/{test-arith2.mc => test-arith2.jl}          |  0
2161  tests/{test-arith3.mc => test-arith3.jl}          |  0
2162  tests/{test-fib.mc => test-fib.jl}                |  0
2163  tests/{test-float1.mc => test-float1.jl}          |  0
2164  tests/{test-float2.mc => test-float2.jl}          |  0
2165  tests/{test-float3.mc => test-float3.jl}          |  0
2166  tests/{test-for1.mc => test-for1.jl}              |  0
2167  tests/{test-for2.mc => test-for2.jl}              |  0
2168  tests/{test-func1.mc => test-func1.jl}            |  0
```

```
 tests/{test-func2.mc => test-func2.jl}           |  0
 tests/{test-func3.mc => test-func3.jl}           |  0
 tests/{test-func4.mc => test-func4.jl}           |  0
 tests/{test-func5.mc => test-func5.jl}           |  0
 tests/{test-func6.mc => test-func6.jl}           |  0
 tests/{test-func7.mc => test-func7.jl}           |  0
 tests/{test-func8.mc => test-func8.jl}           |  0
 tests/{test-func9.mc => test-func9.jl}           |  0
 tests/{test-gcd.mc => test-gcd.jl}               |  0
 tests/{test-gcd2.mc => test-gcd2.jl}             |  0
 tests/{test-global1.mc => test-global1.jl}       |  0
 tests/{test-global2.mc => test-global2.jl}       |  0
 tests/{test-global3.mc => test-global3.jl}       |  0
 tests/{test-hello.mc => test-hello.jl}           |  0
 tests/{test-helloworld.mc => test-helloworld.jl} |  0
 tests/{test-if1.mc => test-if1.jl}               |  0
 tests/{test-if2.mc => test-if2.jl}               |  0
 tests/{test-if3.mc => test-if3.jl}               |  0
 tests/{test-if4.mc => test-if4.jl}               |  0
 tests/{test-if5.mc => test-if5.jl}               |  0
 tests/{test-if6.mc => test-if6.jl}               |  0
 tests/{test-local1.mc => test-local1.jl}         |  0
 tests/{test-local2.mc => test-local2.jl}         |  0
 tests/{test-ops1.mc => test-ops1.jl}             |  0
 tests/{test-ops2.mc => test-ops2.jl}             |  0
 tests/{test-printbig.mc => test-printbig.jl}     |  0
 tests/{test-var1.mc => test-var1.jl}             |  0
 tests/{test-var2.mc => test-var2.jl}             |  0
 tests/{test-while1.mc => test-while1.jl}         |  0
 tests/{test-while2.mc => test-while2.jl}         |  0
 84 files changed, 30 insertions(+), 30 deletions(-)

commit cd9a779b7d1be4d0d82f0d2a36319ac9a073aa86
Author: Frances Cao <frances.te.cao@gmail.com>
Date:   Sun Mar 21 14:08:34 2021 -0400

    add microc template, hello-world deliverable

 Dockerfile                |  32 ++++++
 Makefile                  |  61 ++++++++++++
 README                    |   3 +
 _tags                     |   9 ++
 arcade-font.pbm           | Bin 0 -> 344 bytes
 ast.ml                    | 109 ++++++++++++++++++++
 codegen.ml                | 251 ++++++++++++++++++++++++++++++++++++++++++++++
 font2c                    |   9 ++
 microc.ml                 |  32 ++++++
 microcparse.mly           | 117 ++++++++++++++++++++++
 printbig.c                |  75 +++++++++++++++
 printbig.o                | Bin 0 -> 2080 bytes
 sast.ml                   |  79 +++++++++++++++
 scanner.mll               |  59 ++++++++++++
 semant.ml                 | 190 ++++++++++++++++++++++++++++++++++++++
 testall.sh                | 198 ++++++++++++++++++++++++++++++++++++++
 tests/fail-assign1.err    |   1 +
 tests/fail-assign1.mc     |  11 ++
 tests/fail-assign2.err    |   1 +
```

```
 tests/fail-assign2.mc     |   7 ++
 tests/fail-assign3.err    |   1 +
 tests/fail-assign3.mc     |  11 ++
 tests/fail-dead1.err      |   1 +
 tests/fail-dead1.mc       |   8 ++
 tests/fail-dead2.err      |   1 +
 tests/fail-dead2.mc       |  10 ++
 tests/fail-expr1.err      |   1 +
 tests/fail-expr1.mc       |  18 ++++
 tests/fail-expr2.err      |   1 +
 tests/fail-expr2.mc       |  14 +++
 tests/fail-expr3.err      |   1 +
 tests/fail-expr3.mc       |  14 +++
 tests/fail-float1.err     |   1 +
 tests/fail-float1.mc      |   5 +
 tests/fail-float2.err     |   1 +
 tests/fail-float2.mc      |   5 +
 tests/fail-for1.err       |   1 +
 tests/fail-for1.mc        |  13 +++
 tests/fail-for2.err       |   1 +
 tests/fail-for2.mc        |   8 ++
 tests/fail-for3.err       |   1 +
 tests/fail-for3.mc        |   8 ++
 tests/fail-for4.err       |   1 +
 tests/fail-for4.mc        |   8 ++
 tests/fail-for5.err       |   1 +
 tests/fail-for5.mc        |  10 ++
 tests/fail-func1.err      |   1 +
 tests/fail-func1.mc       |  12 +++
 tests/fail-func2.err      |   1 +
 tests/fail-func2.mc       |   8 ++
 tests/fail-func3.err      |   1 +
 tests/fail-func3.mc       |   8 ++
 tests/fail-func4.err      |   1 +
 tests/fail-func4.mc       |  12 +++
 tests/fail-func5.err      |   1 +
 tests/fail-func5.mc       |  14 +++
 tests/fail-func6.err      |   1 +
 tests/fail-func6.mc       |   9 ++
 tests/fail-func7.err      |   1 +
 tests/fail-func7.mc       |   9 ++
 tests/fail-func8.err      |   1 +
 tests/fail-func8.mc       |  13 +++
 tests/fail-func9.err      |   1 +
 tests/fail-func9.mc       |   9 ++
 tests/fail-global1.err    |   1 +
 tests/fail-global1.mc     |   9 ++
 tests/fail-global2.err    |   1 +
 tests/fail-global2.mc     |   9 ++
 tests/fail-if1.err        |   1 +
 tests/fail-if1.mc         |   6 ++
 tests/fail-if2.err        |   1 +
 tests/fail-if2.mc         |   6 ++
 tests/fail-if3.err        |   1 +
 tests/fail-if3.mc         |   8 ++
 tests/fail-nomain.err     |   1 +
 tests/fail-nomain.mc      |   0
```

```
 tests/fail-print.err      |   1 +
 tests/fail-print.mc       |   2 +
 tests/fail-printb.err     |   1 +
 tests/fail-printb.mc      |   2 +
 tests/fail-printbig.err   |   1 +
 tests/fail-printbig.mc    |   2 +
 tests/fail-return1.err    |   1 +
 tests/fail-return1.mc     |   4 +
 tests/fail-return2.err    |   1 +
 tests/fail-return2.mc     |  10 ++
 tests/fail-while1.err     |   1 +
 tests/fail-while1.mc      |  13 +++
 tests/fail-while2.err     |   1 +
 tests/fail-while2.mc      |  13 +++
 tests/test-add1.mc        |  10 ++
 tests/test-add1.out       |   1 +
 tests/test-arith1.mc      |   5 +
 tests/test-arith1.out     |   1 +
 tests/test-arith2.mc      |   5 +
 tests/test-arith2.out     |   1 +
 tests/test-arith3.mc      |  13 +++
 tests/test-arith3.out     |   1 +
 tests/test-fib.mc         |  16 +++
 tests/test-fib.out        |   6 ++
 tests/test-float1.mc      |   7 ++
 tests/test-float1.out     |   1 +
 tests/test-float2.mc      |  11 ++
 tests/test-float2.out     |   1 +
 tests/test-float3.mc      |  30 ++++++
 tests/test-float3.out     |  24 +++++
 tests/test-for1.mc        |   9 ++
 tests/test-for1.out       |   6 ++
 tests/test-for2.mc        |  11 ++
 tests/test-for2.out       |   6 ++
 tests/test-func1.mc       |  12 +++
 tests/test-func1.out      |   1 +
 tests/test-func2.mc       |  18 ++++
 tests/test-func2.out      |   1 +
 tests/test-func3.mc       |  13 +++
 tests/test-func3.out      |   4 +
 tests/test-func4.mc       |  14 +++
 tests/test-func4.out      |   1 +
 tests/test-func5.mc       |   9 ++
 tests/test-func5.out      |   0
 tests/test-func6.mc       |   9 ++
 tests/test-func6.out      |   1 +
 tests/test-func7.mc       |  13 +++
 tests/test-func7.out      |   1 +
 tests/test-func8.mc       |  10 ++
 tests/test-func8.out      |   1 +
 tests/test-func9.mc       |  11 ++
 tests/test-func9.out      |   1 +
 tests/test-gcd.mc         |  15 +++
 tests/test-gcd.out        |   3 +
 tests/test-gcd2.mc        |  14 +++
 tests/test-gcd2.out       |   3 +
 tests/test-global1.mc     |  30 ++++++
```

```
 tests/test-global1.out    |   4 +
 tests/test-global2.mc     |  10 ++
 tests/test-global2.out    |   1 +
 tests/test-global3.mc     |  11 ++
 tests/test-global3.out    |   1 +
 tests/test-hello.mc       |   7 ++
 tests/test-hello.out      |   3 +
 tests/test-helloworld.mc  |   4 +
 tests/test-helloworld.out |   1 +
 tests/test-if1.mc         |   6 ++
 tests/test-if1.out        |   2 +
 tests/test-if2.mc         |   6 ++
 tests/test-if2.out        |   2 +
 tests/test-if3.mc         |   6 ++
 tests/test-if3.out        |   1 +
 tests/test-if4.mc         |   6 ++
 tests/test-if4.out        |   2 +
 tests/test-if5.mc         |  16 +++
 tests/test-if5.out        |   2 +
 tests/test-if6.mc         |  18 ++++
 tests/test-if6.out        |   2 +
 tests/test-local1.mc      |  13 +++
 tests/test-local1.out     |   1 +
 tests/test-local2.mc      |  14 +++
 tests/test-local2.out     |   1 +
 tests/test-ops1.mc        |  28 ++++++
 tests/test-ops1.out       |  24 +++++
 tests/test-ops2.mc        |  17 ++++
 tests/test-ops2.out       |  14 +++
 tests/test-printbig.mc    |  25 +++++
 tests/test-printbig.out   |  88 ++++++++++++++++++
 tests/test-var1.mc        |   7 ++
 tests/test-var1.out       |   1 +
 tests/test-var2.mc        |  13 +++
 tests/test-var2.out       |   1 +
 tests/test-while1.mc      |  11 ++
 tests/test-while1.out     |   6 ++
 tests/test-while2.mc      |  16 +++
 tests/test-while2.out     |   1 +
 172 files changed, 2331 insertions(+)
```