To launch our GUI for testing use,

py app.py

[More info about how to run our code in README.md]

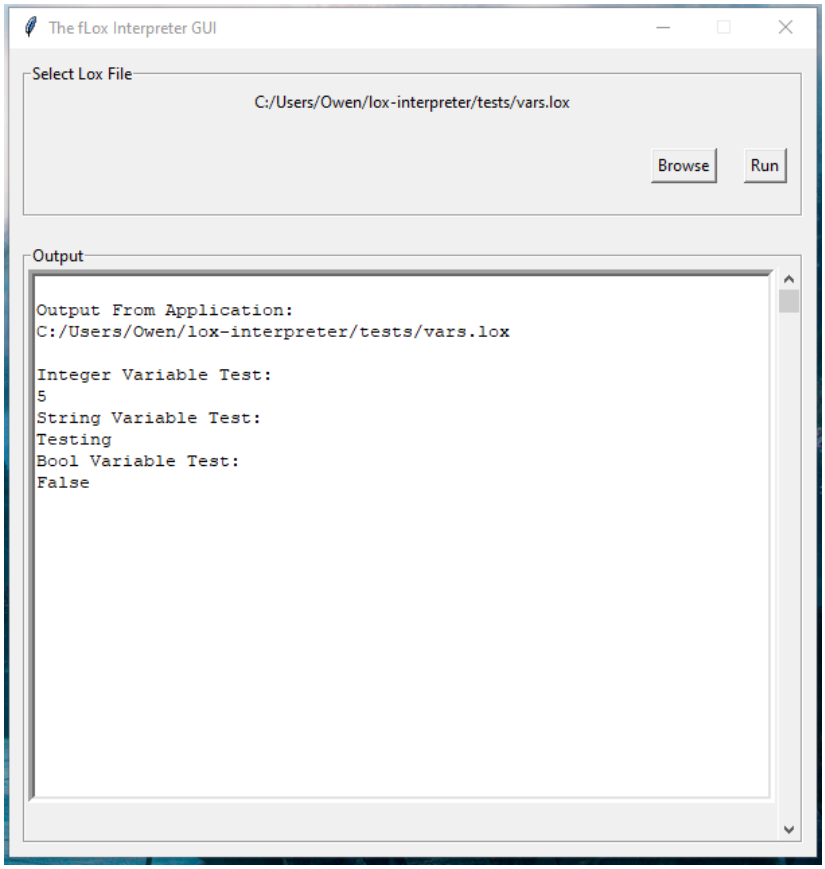
Variable Definition Testing:

Lox Code:

tests > vars.lox

```
1  var testVariable = 5;
2  print "Integer Variable Test:";
3  print testVariable;
4
5  var testVariableString = "Testing";
6  print "String Variable Test:";
7  print testVariableString;
8
9  var testVariableBool = false;
10 print "Bool Variable Test:";
11 print testVariableBool;
```

Output:



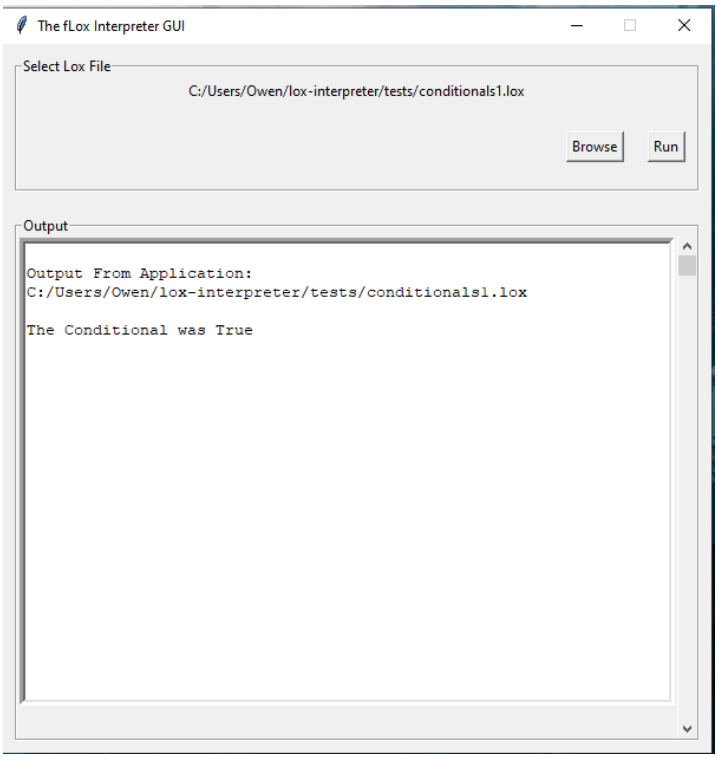
Conditional Testing:

Code:

tests > conditionals1.lox

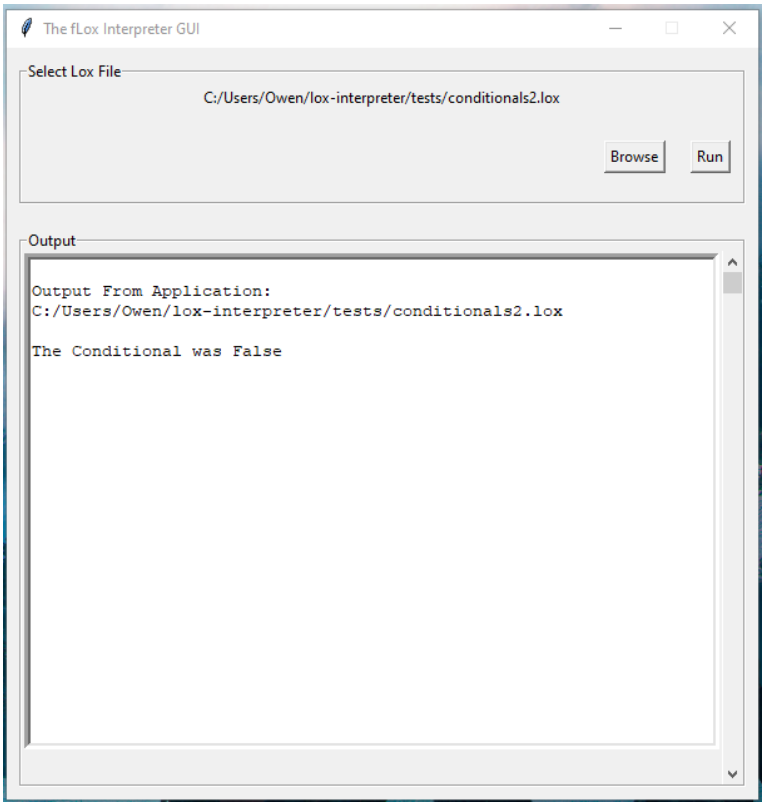
```
var cond = true;
if(cond){
    print "The Conditional was True";
}else{
    print "The Conditional was False";
}

```

Output:



tests > conditionals2.lox

```
var cond = false;
if(cond){
    print "The Conditional was True";
}else{
    print "The Conditional was False";
}

```



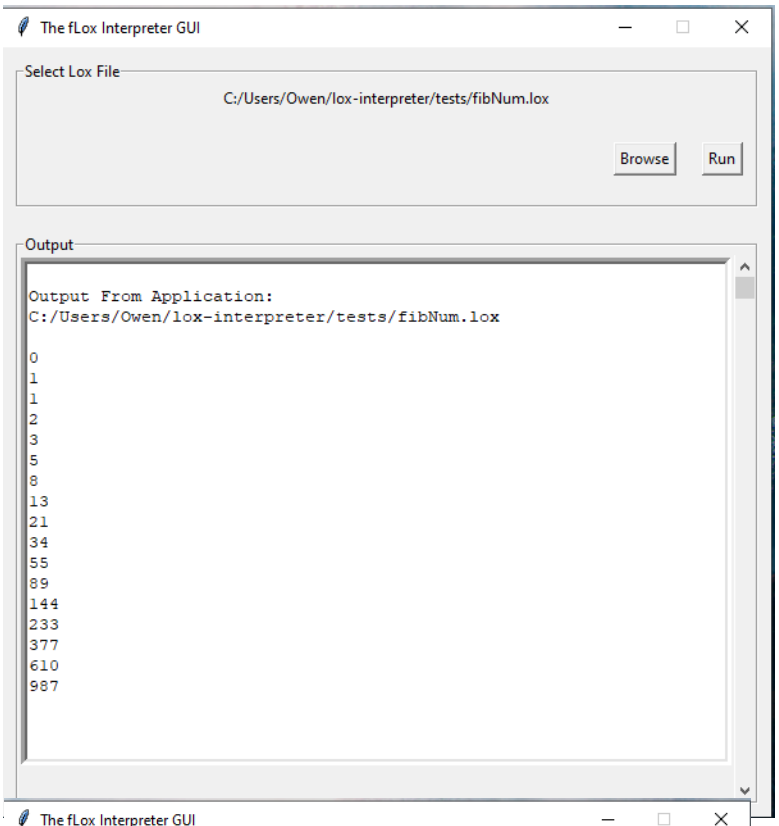
For Loop + While Loop Testing:

Code:

tests > fibNum.lox

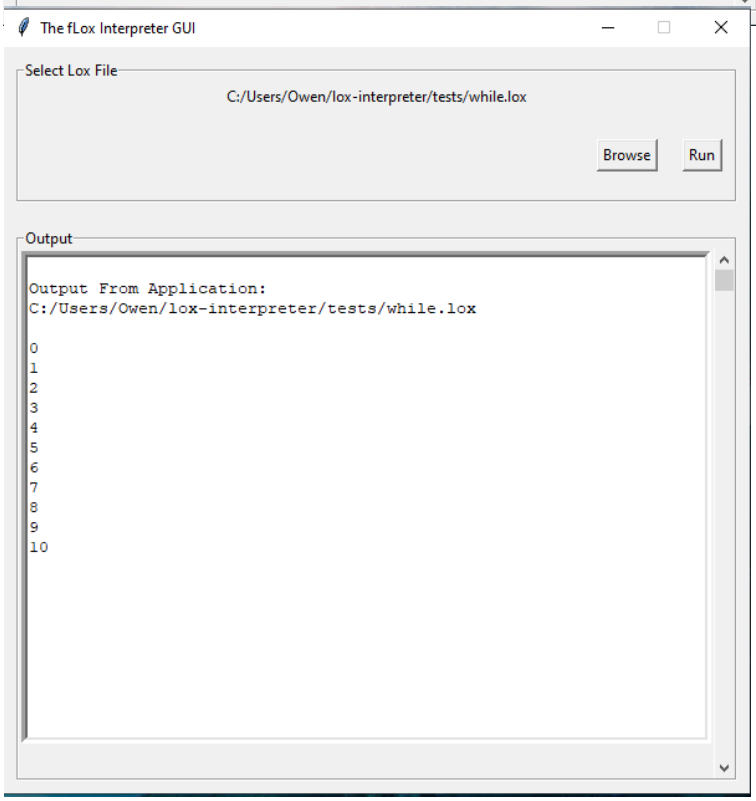
```
var A = 0;
var temp;

for(var B = 1; A < 1000; B = temp+B){
    print A;
    temp = A;
    A = B;
}
```

Output:



tests > while.lox

```
var i = 0;
while(i <= 10){
    print i;
    i = i + 1;
}
```



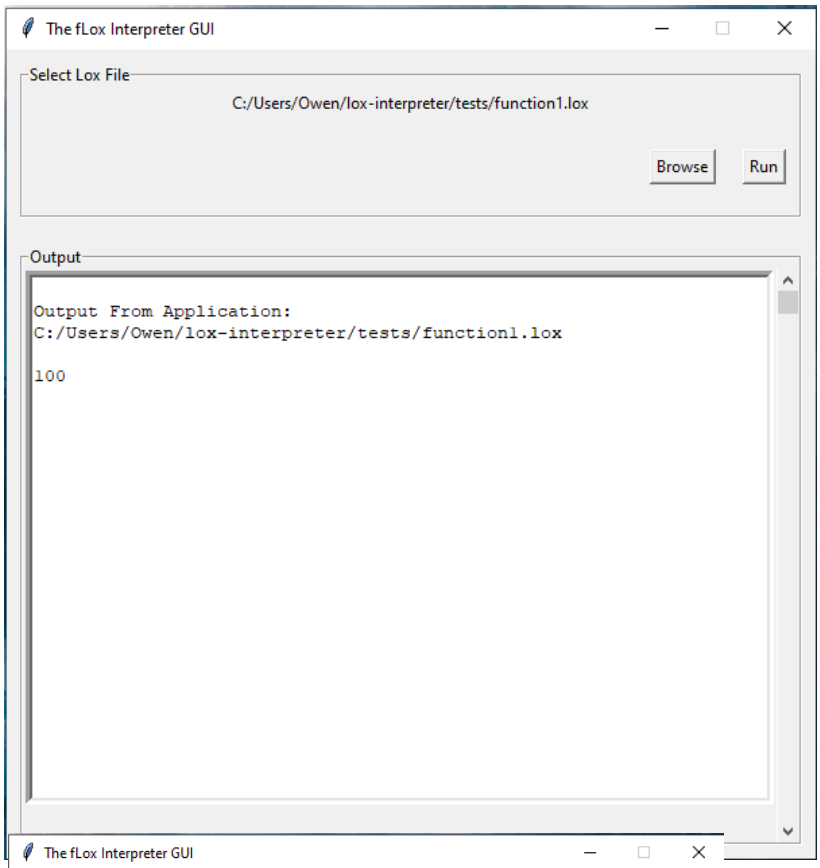
Function Testing:

Code:

tests > function1.lox

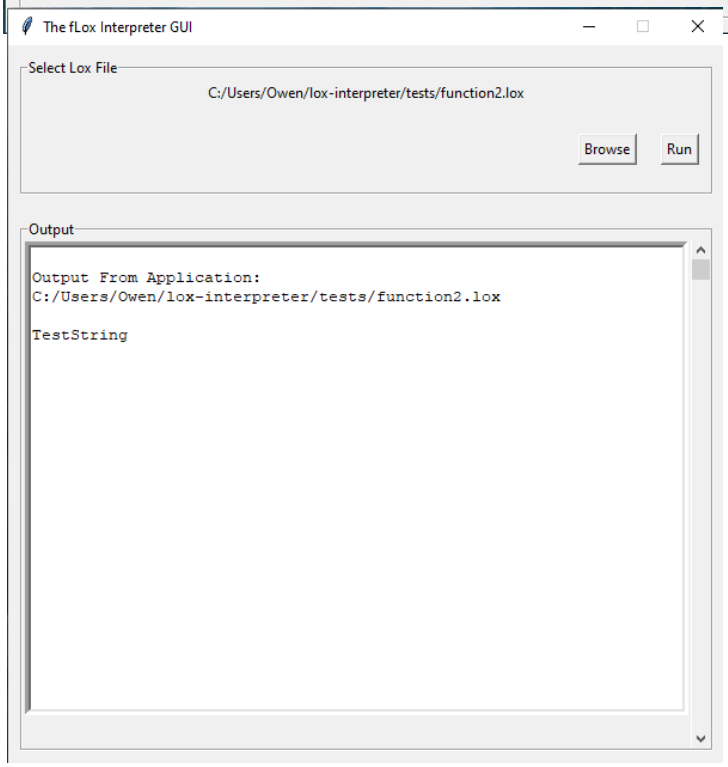
```
1  fun functionTest(integer1, integer2){
2      var answer = integer1 * integer2;
3      print answer;
4  }
5
6  functionTest(10,10);
```

Output:



tests > function2.lox

```
1  fun functionTest2(string1, string2){
2      var concat = string1 + string2;
3      print concat;
4  }
5
6  functionTest2("Test", "String");
7
```

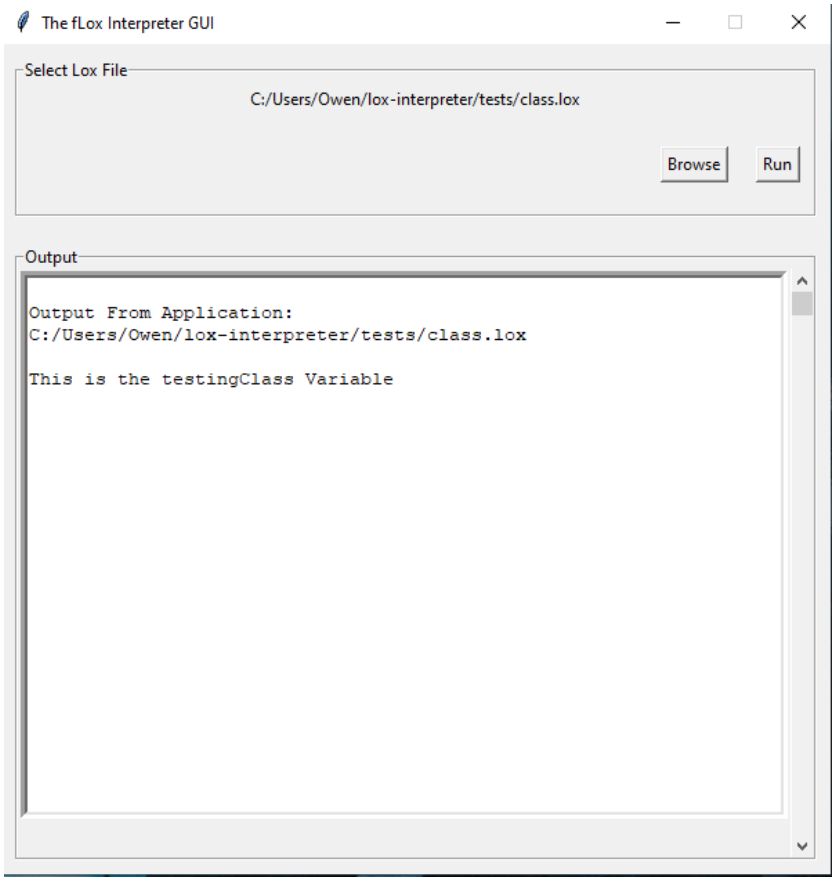


Class Testing:

Code: [file = class.lox in tests folder]

```
//Testing of basic class Initialization
class testingClasses{
    init(){
        var testVar = "This is the testingClass Variable";
        print testVar;
    }
}

testingClasses();
```

Output:

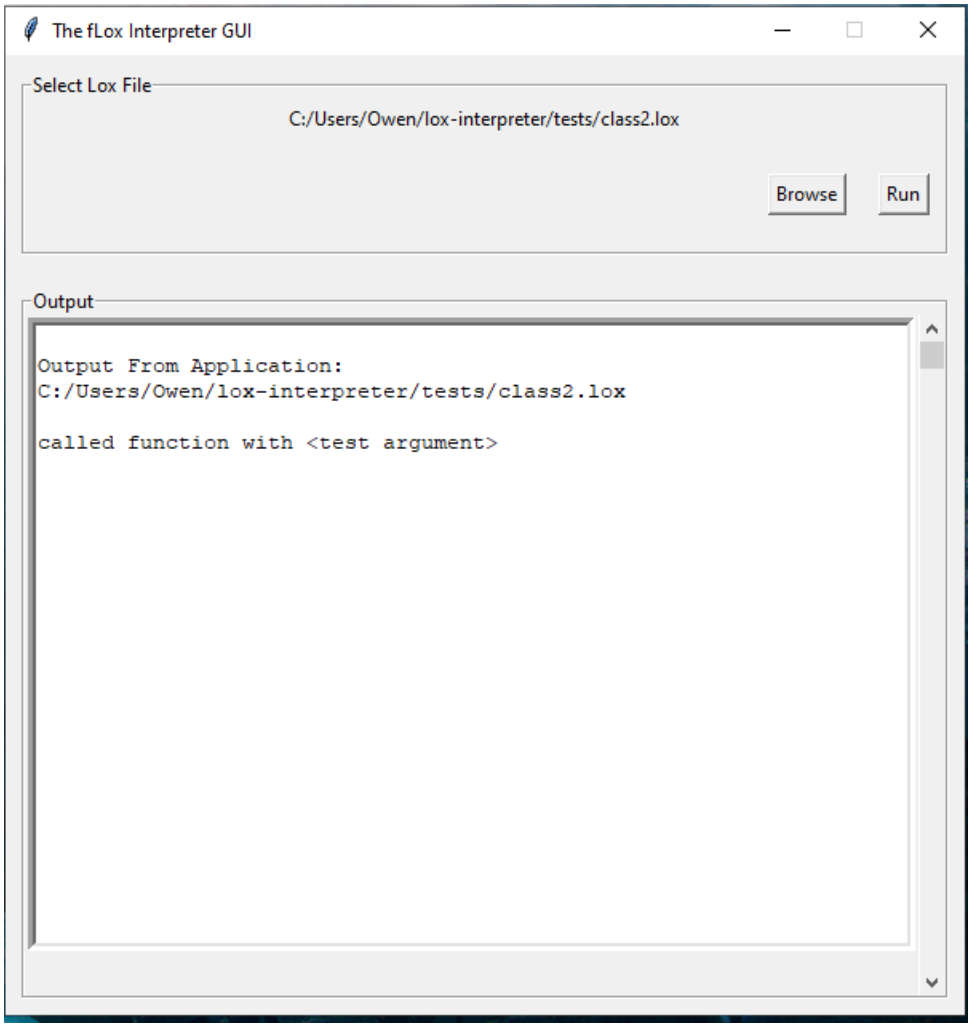


Class Testing:

Code: [file = class2.lox in tests folder]

```
class Box {}

fun notMethod(argument) {
  print "called function with " + argument;
}

var box = Box();
box.function = notMethod;
box.function("<test argument>");
```

Output:

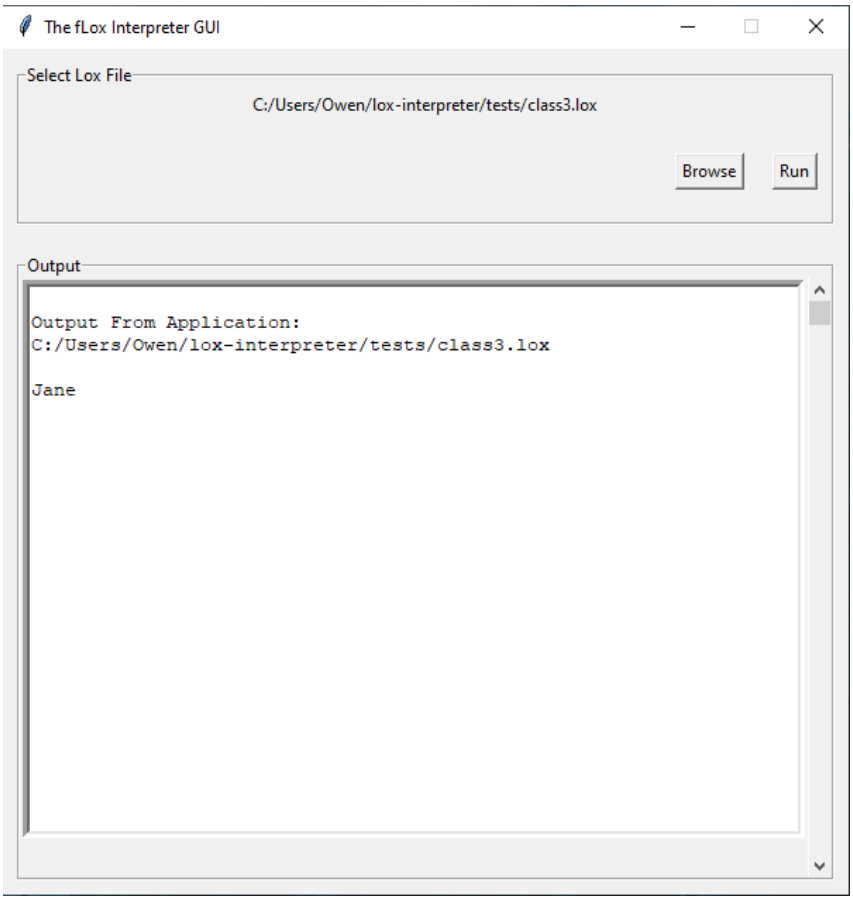


Class Testing:

Code: [file = class3.lox in tests folder]

```
class Person {
  sayName() {
    print this.name;
  }
}

var jane = Person();
jane.name = "Jane";

var bill = Person();
bill.name = "Bill";

bill.sayName = jane.sayName;
bill.sayName(); // Should print Jane
```

Output:

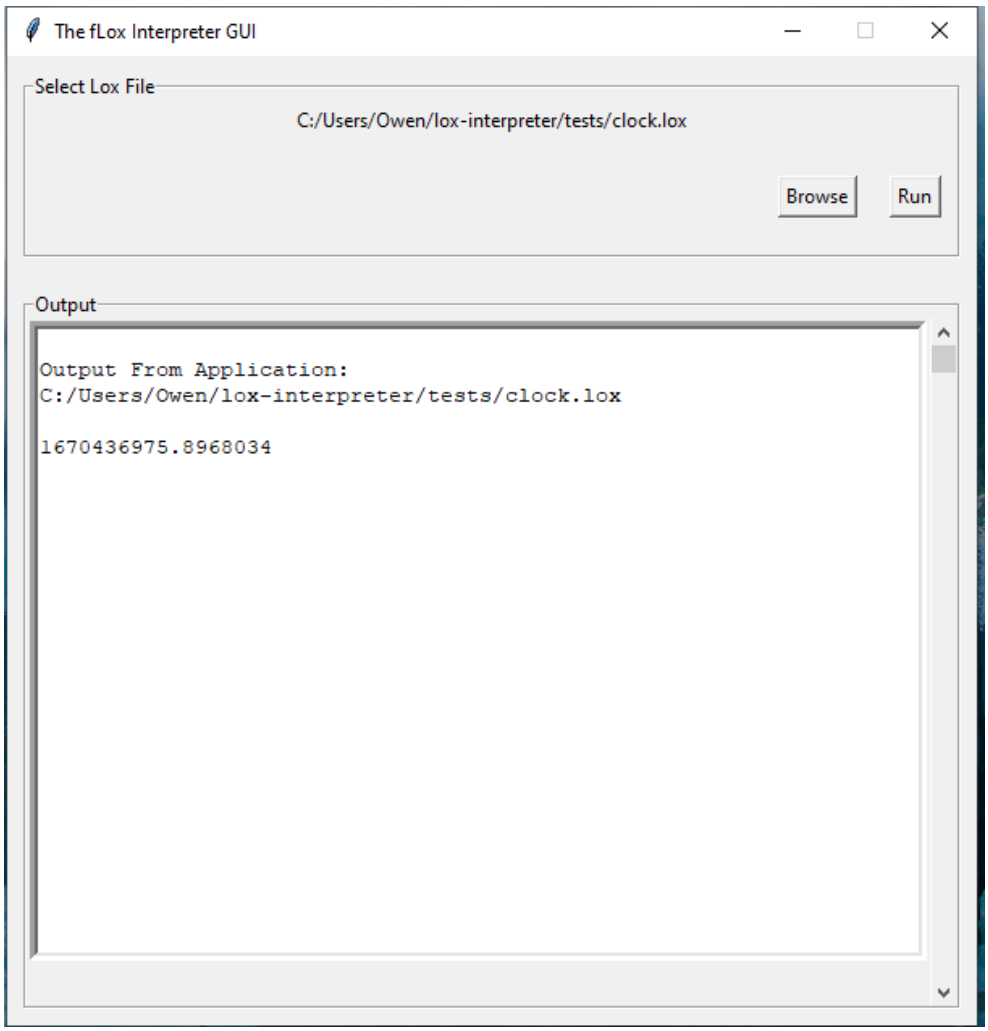


Clock() Test:

Code: [file = clock.lox in tests folder]

```
//Testing of the one built in standard library function of Lox
print clock();
```

Output:

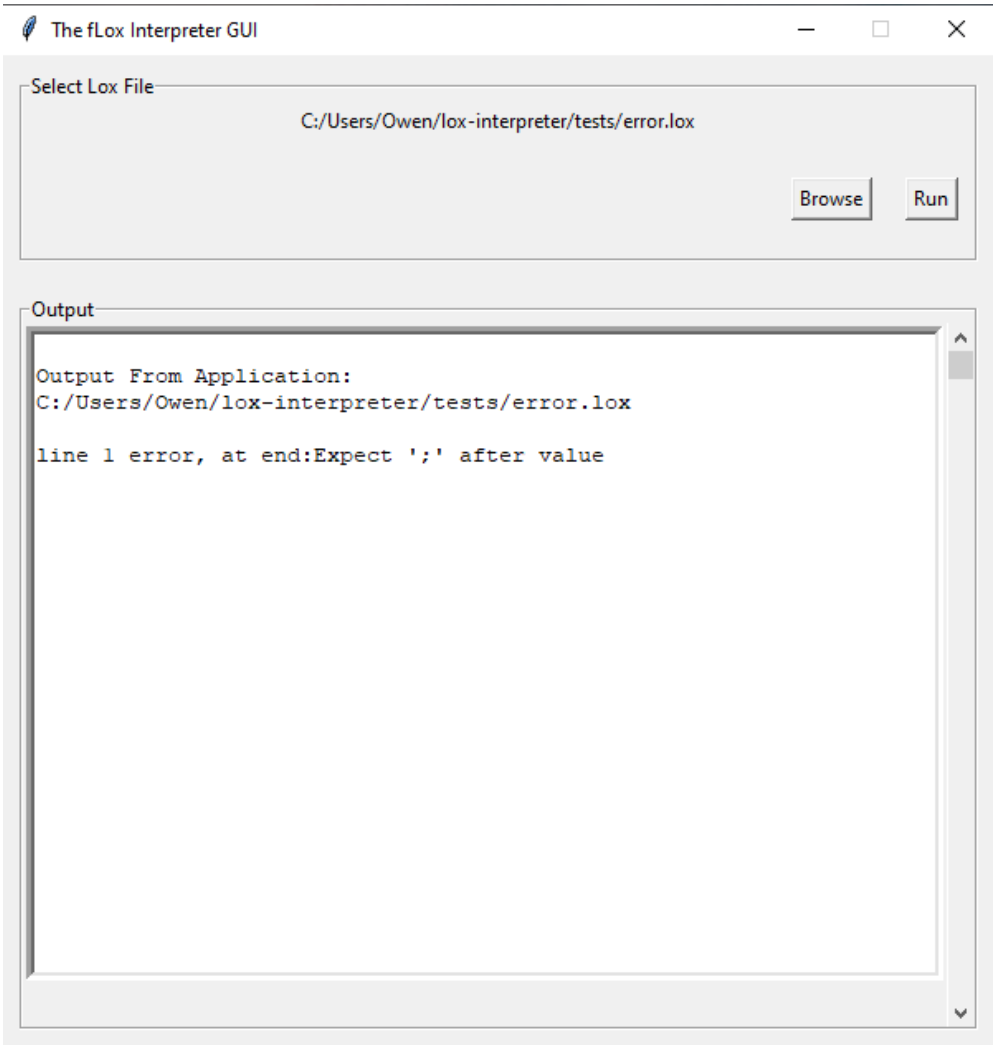


Syntax Error Test:

Code: [file = error.lox in tests folder]

```
print "This breaks"
```

Output:

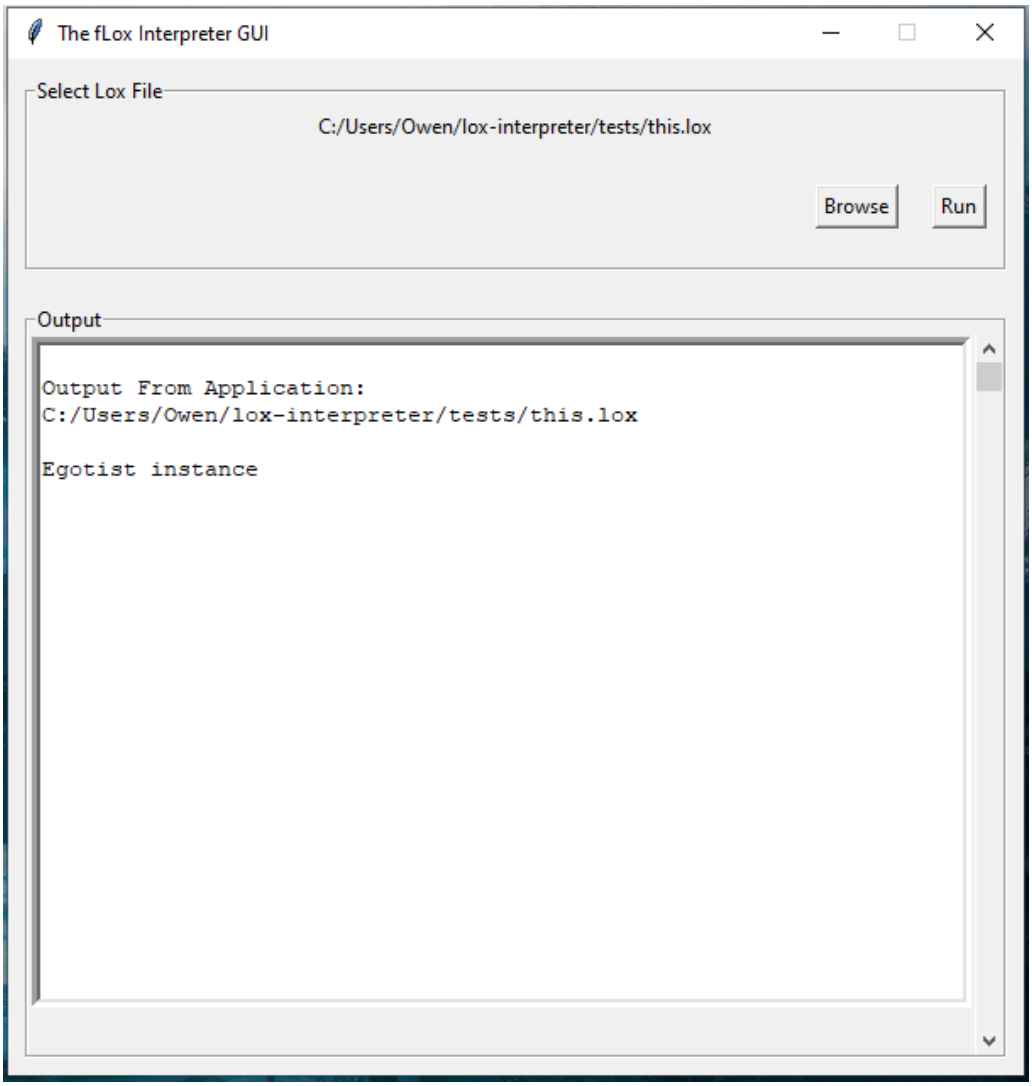


Class [this] Testing:

Code: [file = this.lox in tests folder]

```
class Egotist {
  speak() {
    print this;
  }
}

var method = Egotist().speak;
method();
```

Output:

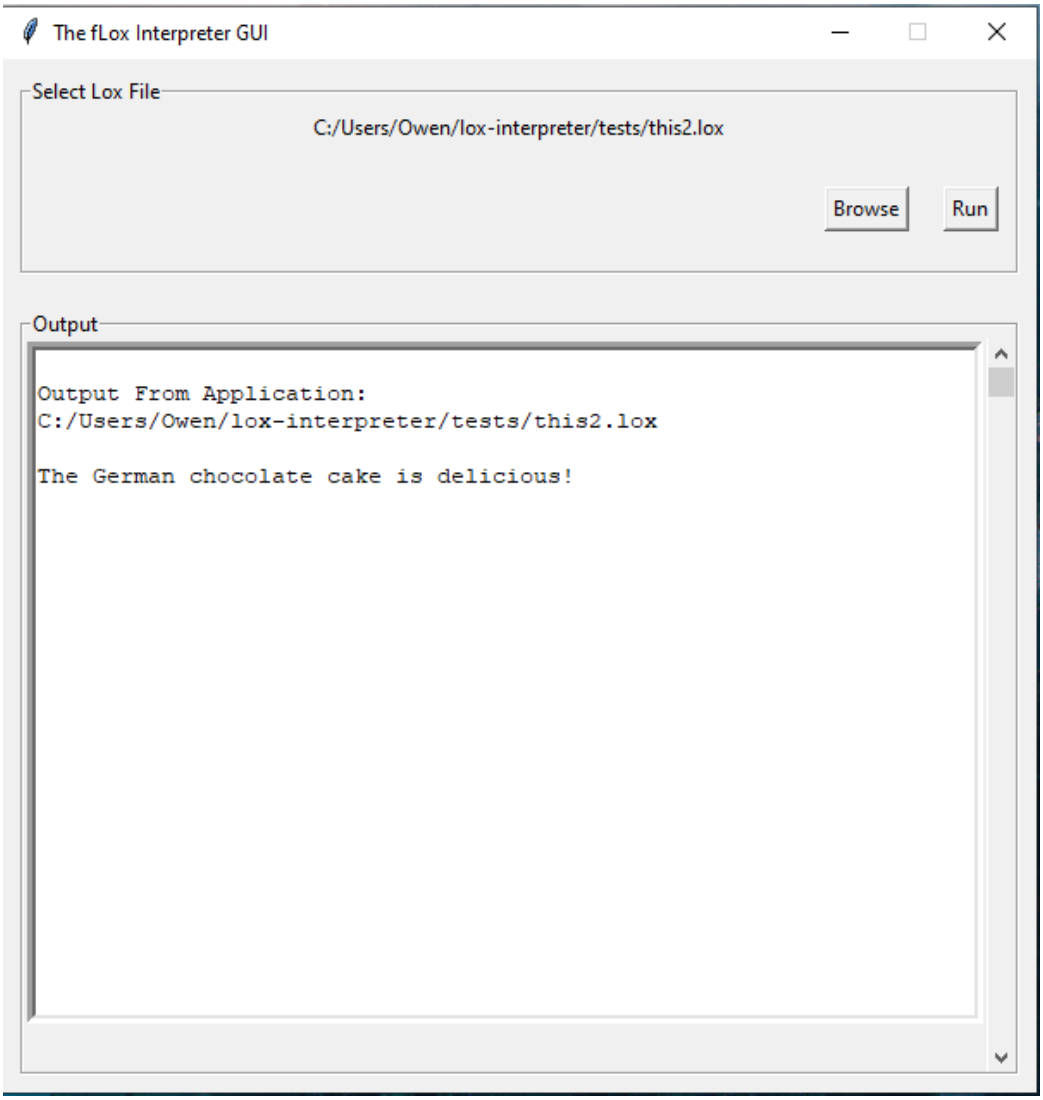


Class [this] Testing:

Code: [file = this2.lox in tests folder]

```
class Cake {
  taste() {
    var adjective = "delicious";
    print "The " + this.flavor + " cake is " + adjective + "!";
  }
}

var cake = Cake();
cake.flavor = "German chocolate";
cake.taste(); // Prints "The German chocolate cake is delicious!".
```

Output:



Inheritance Test:

Code: [file = inheritance.lox in tests folder]

```
class Doughnut {
  cook() {
    print "Fry until golden brown.";
  }
  serve(){
    print "Place in a nice doughnut box!";
  }
}

class BostonCream < Doughnut {
  cook() {
    super.cook();
    print "Pipe full of custard and coat with chocolate.";
    super.serve();
  }
}

BostonCream().cook();
```

Output: