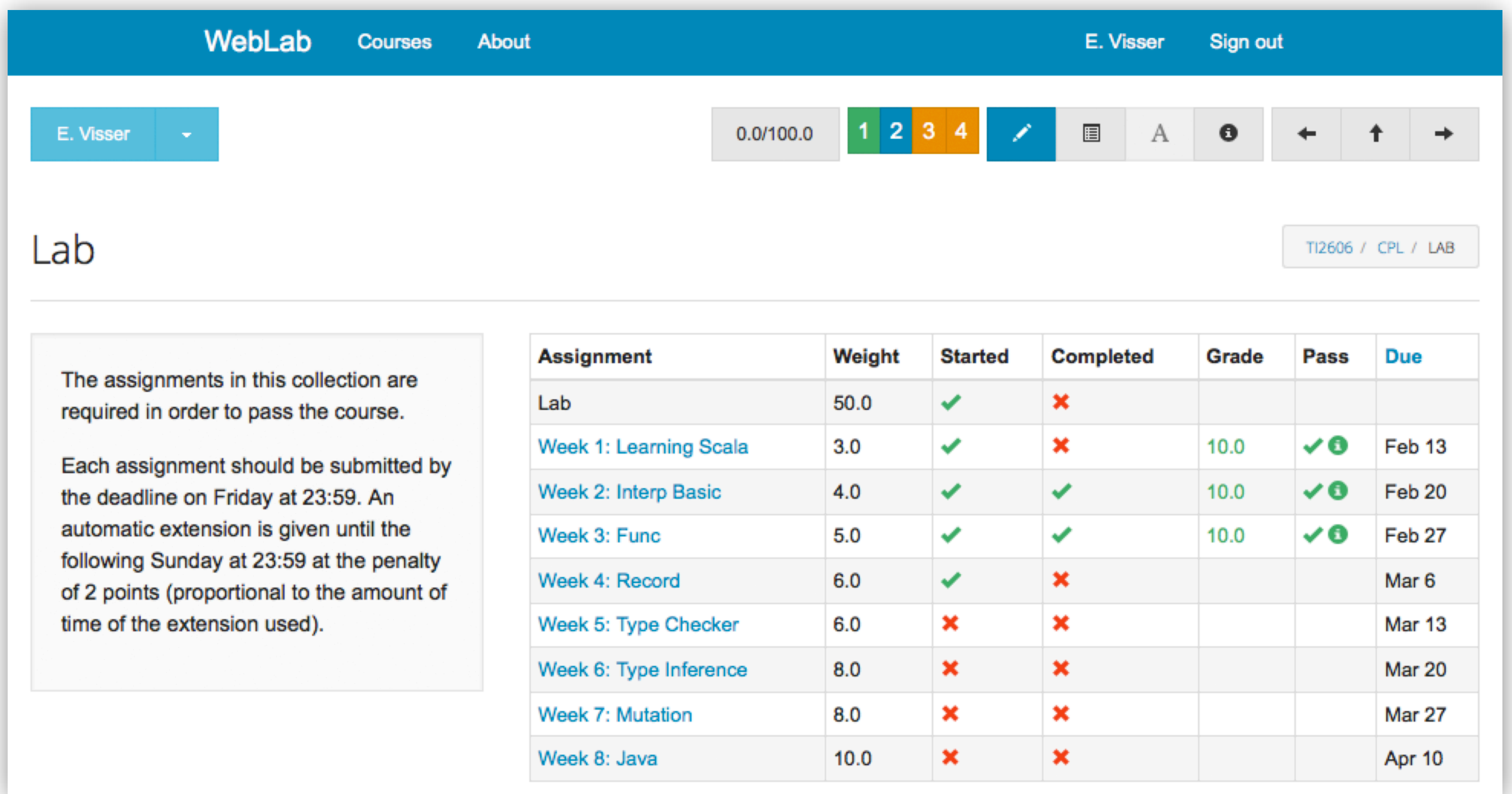
WebLab

Programming Education in the Browser

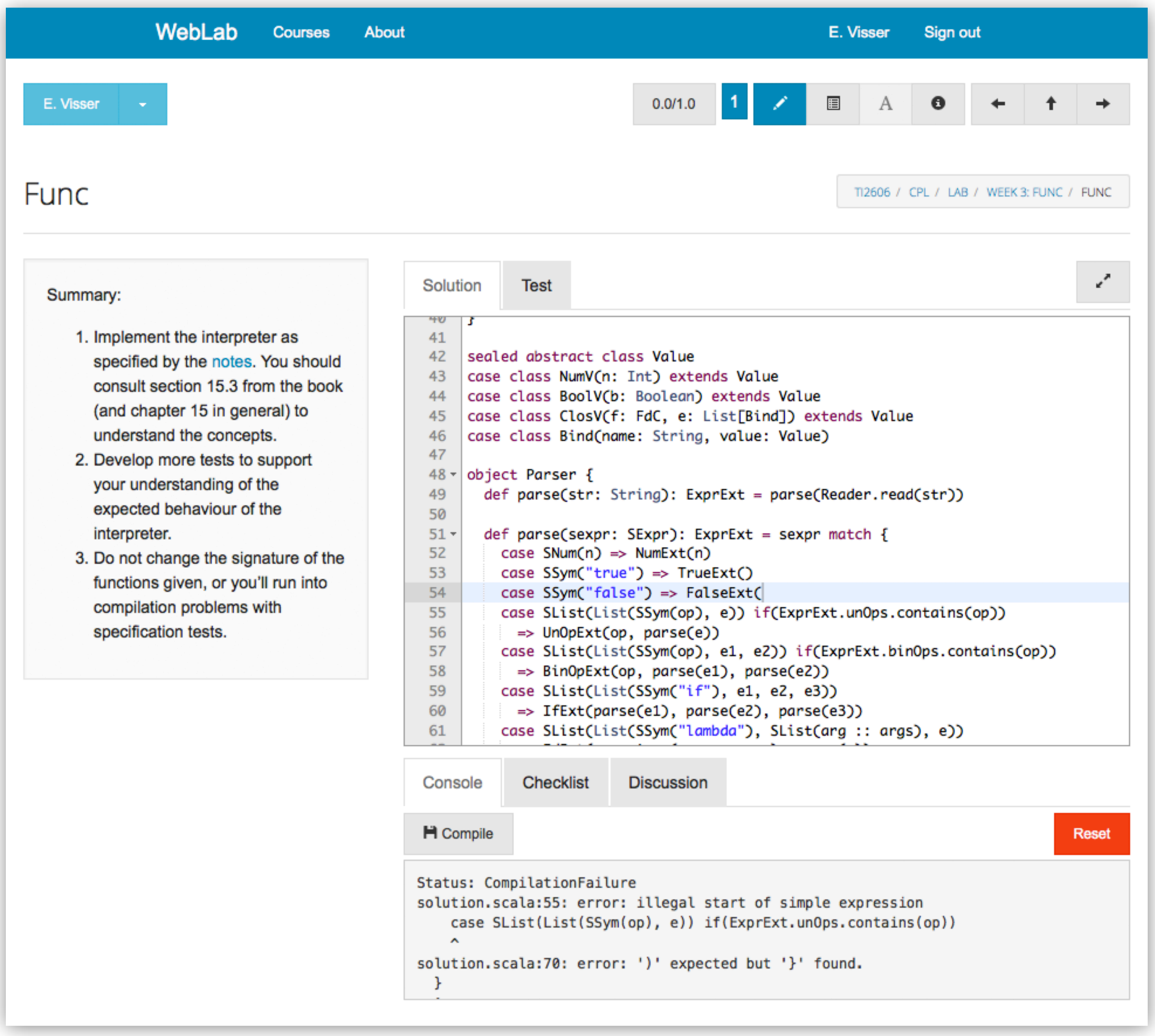


Lab

The assignments in this collection are required in order to pass the course.

Each assignment should be submitted by the deadline on Friday at 23:59. An automatic extension is given until the following Sunday at 23:59 at the penalty of 2 points (proportional to the amount of time of the extension used).

Assignment	Weight	Started	Completed	Grade	Pass	Due
Lab	50.0	✓	✗			
Week 1: Learning Scala	3.0	✓	✗	10.0	✓ ℹ	Feb 13
Week 2: Interp Basic	4.0	✓	✓	10.0	✓ ℹ	Feb 20
Week 3: Func	5.0	✓	✓	10.0	✓ ℹ	Feb 27
Week 4: Record	6.0	✓	✗			Mar 6
Week 5: Type Checker	6.0	✗	✗			Mar 13
Week 6: Type Inference	8.0	✗	✗			Mar 20
Week 7: Mutation	8.0	✗	✗			Mar 27
Week 8: Java	10.0	✗	✗			Apr 10



Func

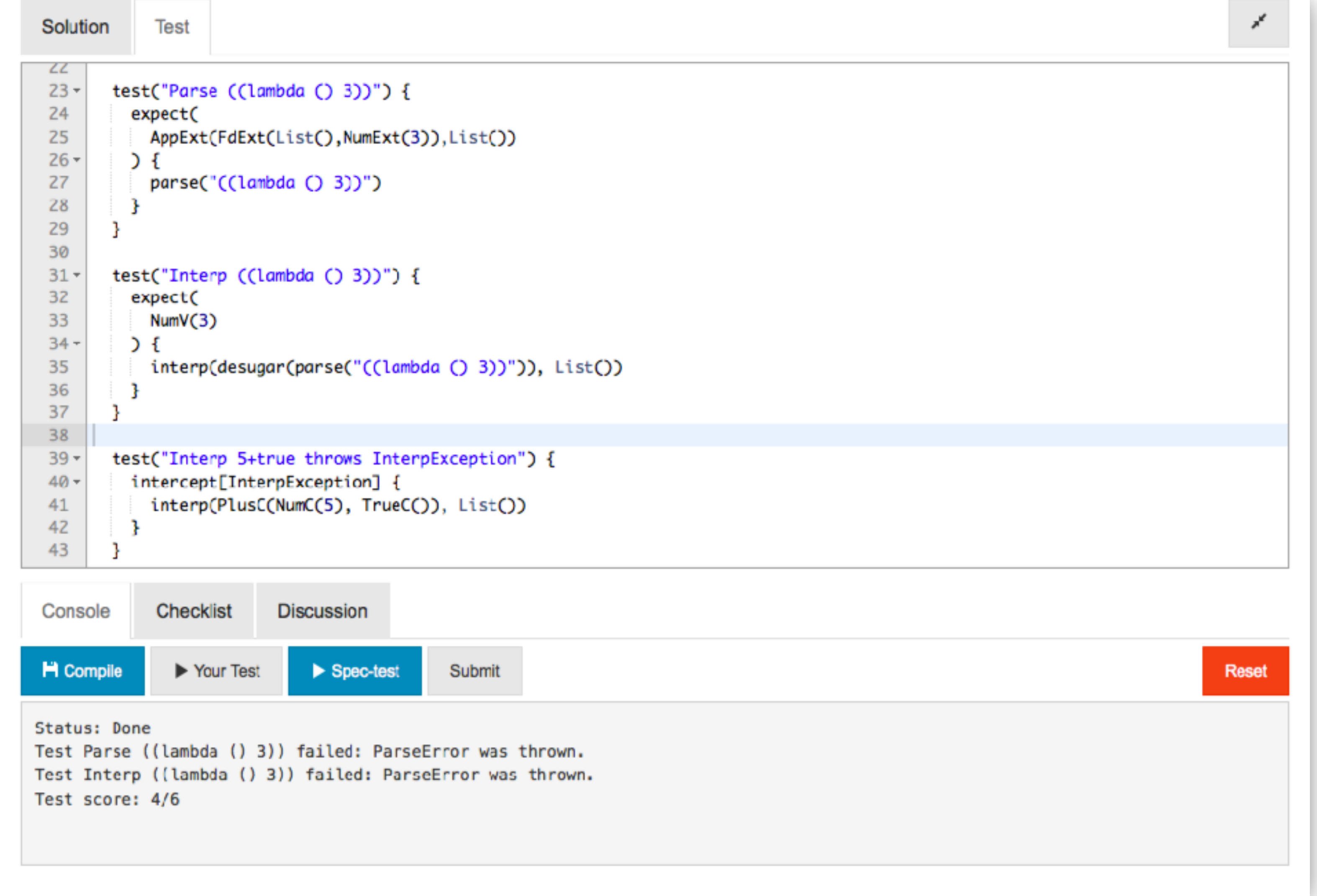
Summary:

1. Implement the interpreter as specified by the notes. You should consult section 15.3 from the book (and chapter 15 in general) to understand the concepts.
2. Develop more tests to support your understanding of the expected behaviour of the interpreter.
3. Do not change the signature of the functions given, or you'll run into compilation problems with specification tests.

```scala
}

sealed abstract class Value
case class NumV(n: Int) extends Value
case class BoolV(b: Boolean) extends Value
case class ClosV(f: FdC, e: List[Bind]) extends Value
case class Bind(name: String, value: Value)

object Parser {
  def parse(str: String): ExprExt = parse(Reader.read(str))

  def parse(sexpr: SExpr): ExprExt = sexpr match {
    case SNum(n) => NumExt(n)
    case SSym("true") => TrueExt()
    case SSym("false") => FalseExt(
    case SList(List(SSym(op), e)) if(ExprExt.unOps.contains(op))
      => UnOpExt(op, parse(e))
    case SList(List(SSym(op), e1, e2)) if(ExprExt.binOps.contains(op))
      => BinOpExt(op, parse(e1), parse(e2))
    case SList(List(SSym("if"), e1, e2, e3))
      => IfExt(parse(e1), parse(e2), parse(e3))
    case SList(List(SSym("lambda"), SList(arg :: args), e))
```

```
Status: CompilationFailure
solution.scala:55: error: illegal start of simple expression
    case SList(List(SSym(op), e)) if(ExprExt.unOps.contains(op))
    ^
solution.scala:70: error: ')' expected but '}' found.
  }
```

Students read assignments and write programming solutions in the browser. Submitted programs are compiled and executed immediately on the server.



```scala
  test("Parse ((lambda () 3))") {
    expect(
      AppExt(FdExt(List(),NumExt(3)),List())
    ) {
      parse("((lambda () 3))")
    }
  }

  test("Interp ((lambda () 3))") {
    expect(
      NumV(3)
    ) {
      interp(desugar(parse("((lambda () 3))")), List())
    }
  }

  test("Interp 5+true throws InterpException") {
    intercept[InterpException] {
      interp(PlusC(NumC(5), TrueC()), List())
    }
  }
```

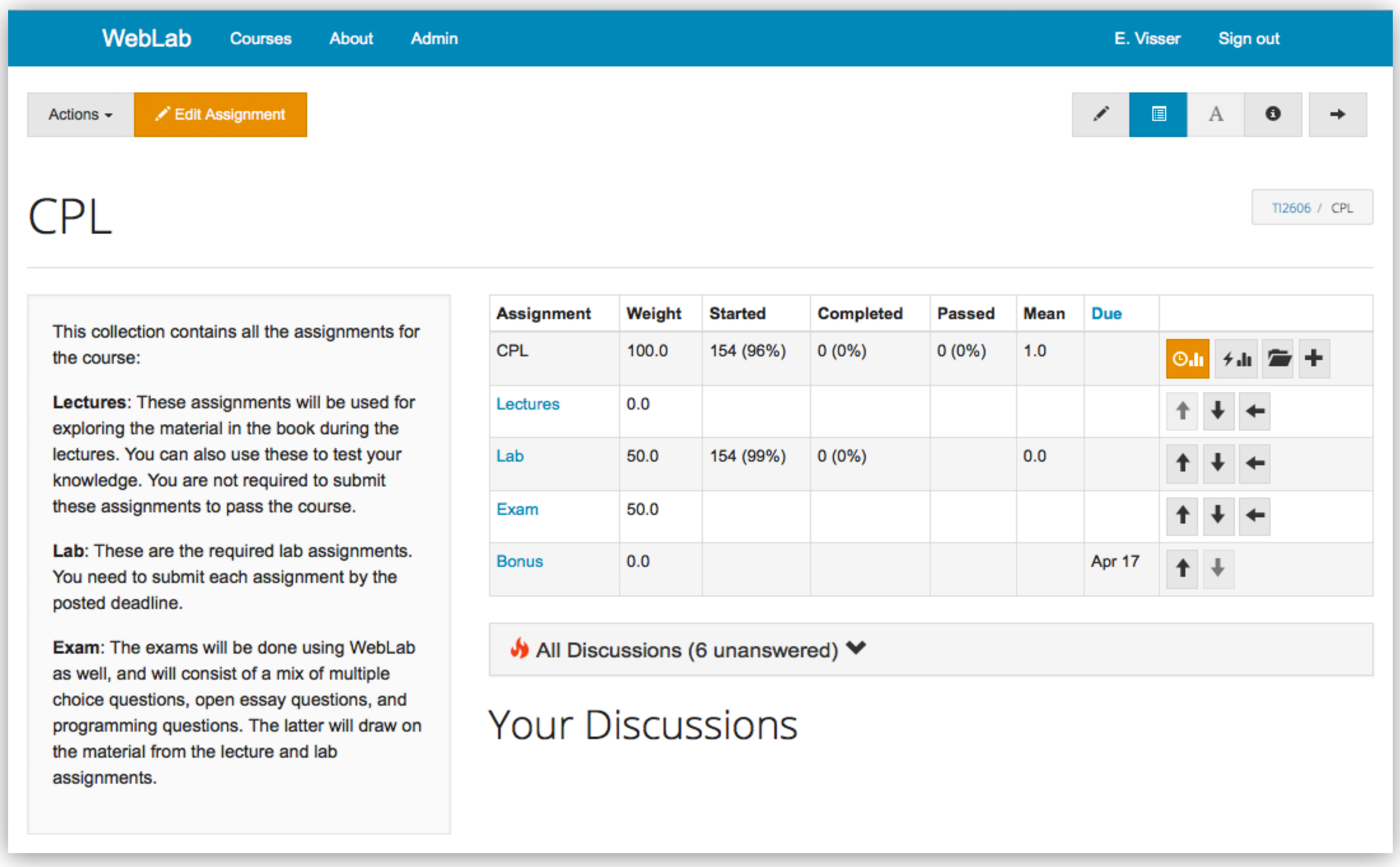
```
Status: Done
Test Parse ((lambda () 3)) failed: ParseError was thrown.
Test Interp ((lambda () 3)) failed: ParseError was thrown.
Test score: 4/6
```

Solutions are checked for correctness using unit tests, which are reported directly to students

http://weblab.tudelft.nl

Traditional programming education has a long feedback cycle with students making assignments in the lab and teaching assistants manually grading (sometimes still on paper) submitted assignments. In addition, the approach requires installation of programming environments on lab machines and/or a wide variety of student's own computers. Finally, exams are typically conducted on paper, making it hard to test programming skills and insight, and again making for a large grading effort.

WebLab is a web-based learning management system with a focus on programming education. The application is provided as a web service that can be used by instructors and students at any computer with an internet connection. WebLab completely avoids the problem of installing programming environments. By adding support for a programming language to WebLab once, it can be used in all courses by all students.



CPL

This collection contains all the assignments for the course:

Lectures: These assignments will be used for exploring the material in the book during the lectures. You can also use these to test your knowledge. You are not required to submit these assignments to pass the course.

Lab: These are the required lab assignments. You need to submit each assignment by the posted deadline.

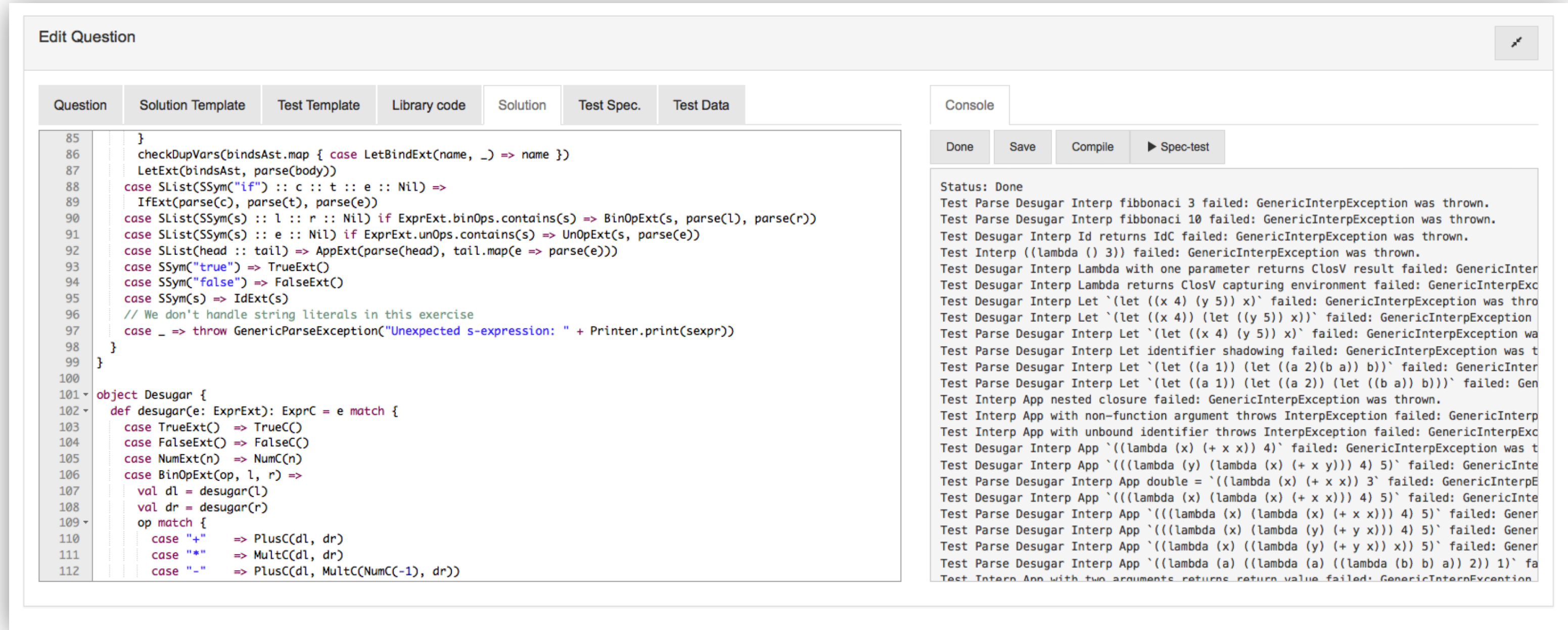
Exam: The exams will be done using WebLab as well, and will consist of a mix of multiple choice questions, open essay questions, and programming questions. The latter will draw on the material from the lecture and lab assignments.

Assignment	Weight	Started	Completed	Passed	Mean	Due
CPL	100.0	154 (96%)	0 (0%)	0 (0%)	1.0	
Lectures	0.0					
Lab	50.0	154 (99%)	0 (0%)		0.0	
Exam	50.0					
Bonus	0.0					Apr 17

All Discussions (6 unanswered)

Your Discussions

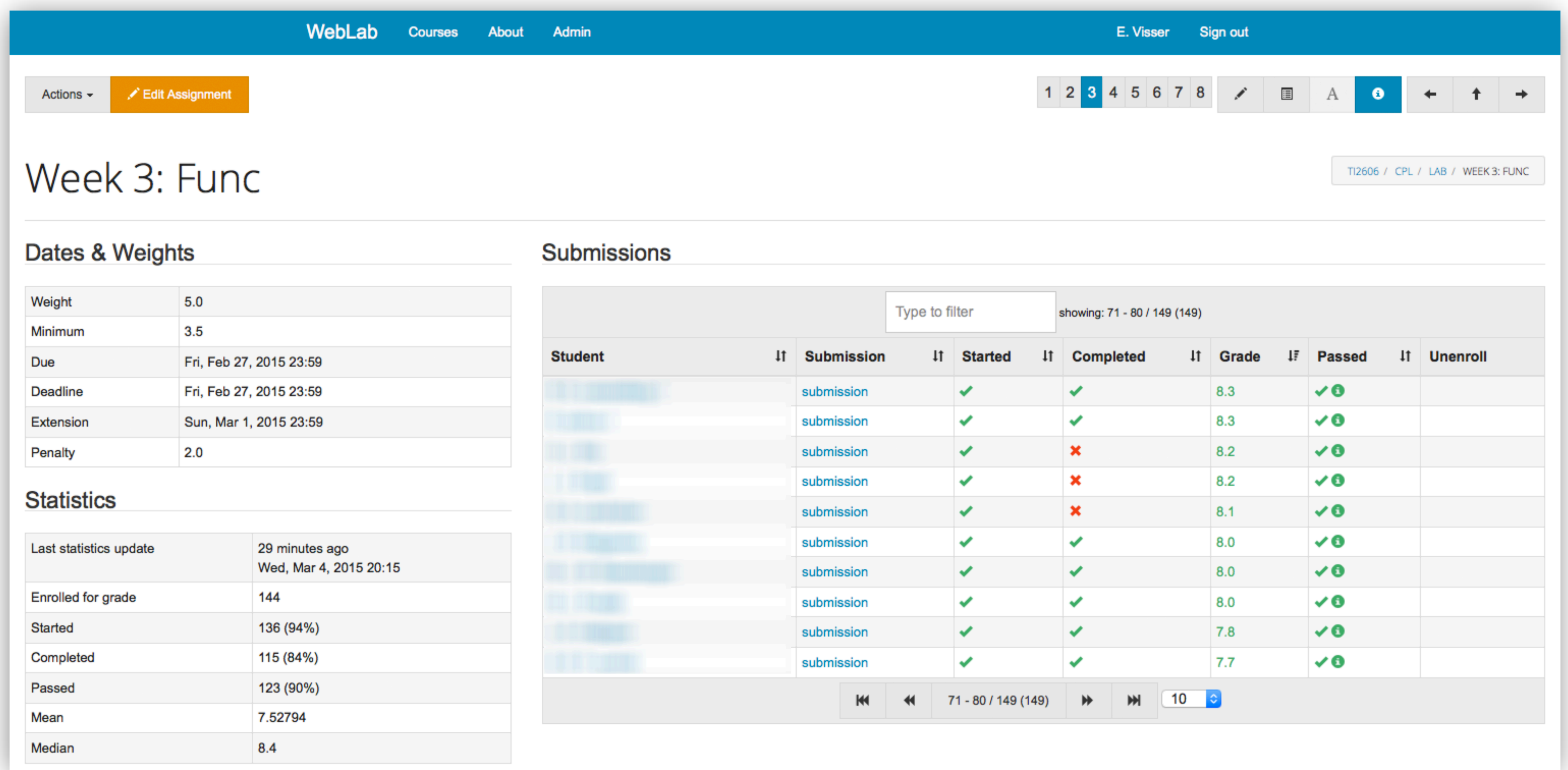
Instructors use the same environment to design the course set-up, develop and test assignments, …



```scala
      }
      checkDupVars(bindsAst.map { case LetBindExt(name, _) => name })
      LetExt(bindsAst, parse(body))
    case SList(SSym("if") :: c :: t :: e :: Nil) =>
      IfExt(parse(c), parse(t), parse(e))
    case SList(SSym(s) :: l :: r :: Nil) if ExprExt.binOps.contains(s) => BinOpExt(s, parse(l), parse(r))
    case SList(SSym(s) :: e :: Nil) if ExprExt.unOps.contains(s) => UnOpExt(s, parse(e))
    case SList(head :: tail) => AppExt(parse(head), tail.map(e => parse(e)))
    case SSym("true") => TrueExt()
    case SSym("false") => FalseExt()
    case SSym(s) => IdExt(s)
    // We don't handle string literals in this exercise
    case _ => throw GenericParseException("Unexpected s-expression: " + Printer.print(sexpr))
  }
}

object Desugar {
  def desugar(e: ExprExt): ExprC = e match {
    case TrueExt()  => TrueC()
    case FalseExt() => FalseC()
    case NumExt(n)  => NumC(n)
    case BinOpExt(op, l, r) =>
      val dl = desugar(l)
      val dr = desugar(r)
      op match {
        case "+"    => PlusC(dl, dr)
        case "*"    => MultC(dl, dr)
        case "-"    => PlusC(dl, MultC(NumC(-1), dr))
```

```
Status: Done
Test Parse Desugar Interp fibbonaci 3 failed: GenericInterpException was thrown.
Test Parse Desugar Interp fibbonaci 10 failed: GenericInterpException was thrown.
Test Desugar Interp Id returns IdC failed: GenericInterpException was thrown.
Test Interp ((lambda () 3)) failed: GenericInterpException was thrown.
Test Desugar Interp Lambda with one parameter returns ClosV result failed: GenericInter
Test Desugar Interp Lambda returns ClosV capturing environment failed: GenericInterpExc
Test Desugar Interp Let `(let ((x 4) (y 5)) x)` failed: GenericInterpException was thro
Test Desugar Interp Let `(let ((x 4)) (let ((y 5)) x))` failed: GenericInterpException
Test Parse Desugar Interp Let `(let ((x 4) (y 5)) x)` failed: GenericInterpException wa
Test Parse Desugar Interp Let identifier shadowing failed: GenericInterpException was t
Test Parse Desugar Interp Let `(let ((a 1)) (let ((a 2)(b a)) b))` failed: GenericInter
Test Parse Desugar Interp Let `(let ((a 1)) (let ((a 2)) (let ((b a)) b)))` failed: Gen
Test Interp App nested closure failed: GenericInterpException was thrown.
Test Interp App with non-function argument throws InterpException failed: GenericInterp
Test Interp App with unbound identifier throws InterpException failed: GenericInterpExc
Test Desugar Interp App `((lambda (x) (+ x x)) 4)` failed: GenericInterpException was t
Test Desugar Interp App `(((lambda (y) (lambda (x) (+ x y))) 4) 5)` failed: GenericInte
Test Parse Desugar Interp App double = `((lambda (x) (+ x x)) 3` failed: GenericInterpE
Test Desugar Interp App `(((lambda (x) (lambda (x) (+ x x))) 4) 5)` failed: GenericInte
Test Parse Desugar Interp App `(((lambda (x) (lambda (x) (+ x x))) 4) 5)` failed: Gener
Test Parse Desugar Interp App `(((lambda (x) (lambda (y) (+ y x))) 4) 5)` failed: Gener
Test Parse Desugar Interp App `((lambda (x) ((lambda (y) (+ y x)) x)) 5)` failed: Gener
Test Parse Desugar Interp App `((lambda (a) ((lambda (a) ((lambda (b) b) a)) 2)) 1)` fa
```

…, monitor student progress, and keep track of course administration



Week 3: Func

Dates & Weights

Weight	5.0
Minimum	3.5
Due	Fri, Feb 27, 2015 23:59
Deadline	Fri, Feb 27, 2015 23:59
Extension	Sun, Mar 1, 2015 23:59
Penalty	2.0

Statistics

Last statistics update	29 minutes ago Wed, Mar 4, 2015 20:15
Enrolled for grade	144
Started	136 (94%)
Completed	115 (84%)
Passed	123 (90%)
Mean	7.52794
Median	8.4

Submissions

showing: 71 - 80 / 149 (149)

Student	Submission	Started	Completed	Grade	Passed	Unenroll
[redacted]	submission	✓	✓	8.3	✓ ℹ	
[redacted]	submission	✓	✓	8.3	✓ ℹ	
[redacted]	submission	✓	✗	8.2	✓ ℹ	
[redacted]	submission	✓	✗	8.2	✓ ℹ	
[redacted]	submission	✓	✗	8.1	✓ ℹ	
[redacted]	submission	✓	✓	8.0	✓ ℹ	
[redacted]	submission	✓	✓	8.0	✓ ℹ	
[redacted]	submission	✓	✓	8.0	✓ ℹ	
[redacted]	submission	✓	✓	7.8	✓ ℹ	
[redacted]	submission	✓	✓	7.7	✓ ℹ	

71 - 80 / 149 (149)   10

WebLab assignments can be used during scheduled **lab sessions**, so that teaching assistants can focus on explaining rather than checking solutions. WebLab can also be used to administer **computer-based** programming **exams**. Students sign in to the exam using a personal key handed out on paper to verify physical presence in the exam room. Using unit-testing based grading, exam grades can be published on the day of the exam.

TU Delft Delft University of Technology

Elmer van Chastelet I Vlad Vergu I Eelco Visser

{e.vanchastelet,v.a.vergu,e.visser}@tudelft.nl

DELFT DATA SCIENCE