

# EE360T/382V Software Testing

khurshid@ece.utexas.edu

April 25, 2018

# Overview

Last time – Symbolic execution

- Testing multi-threaded code
- Testing websites

Today – Review for Exam 3

Next time – **Exam 3 on Monday, April 30**

- In-class, closed book
- Focus: Practical considerations (Text), systematic testing, non-det. choice, symbolic execution

Problem Set 6 – Due: 5/4 11:59pm

Practice for Exam 3, i.e., Problem Set 7 is out – Ungraded

# EE360T/382V Software Testing

khurshid@ece.utexas.edu

## Practical considerations (Chapter 6)\*

\*Introduction to Software Testing by Ammann and Offutt

# Unit testing object-oriented (OO) code

Specifications for OO code have three key components:

- **Class invariants** (e.g., *repOk* methods)
- **Method preconditions and postconditions**

Testing a method requires creating its inputs – **pre-state**

- For instance method: receiver object and arguments

Two basic approaches to input generation

- Abstract level—use method sequences
- Representation level—allocate objects and set fields
  - Can combine the two approaches

# Symbolic execution

Technique for executing the program on **symbolic** input values to analyze its behaviors

- Invented in 70's; heavily studied in last 15 years
- Explores **symbolic execution tree** of (bounded) paths
  - For each path, builds a **path condition** that represents inputs that execute that path
  - Checks satisfiability of path conditions to try to avoid infeasible paths

Program state: symbolic values of variables, path condition, and counter

Various applications, e.g., for test input generation

Traditional focus: programs with primitives and arrays

# Selenium [[www.seleniumhq.org](http://www.seleniumhq.org)]



**Selenium is a suite of tools** to automate web browsers across many platforms.

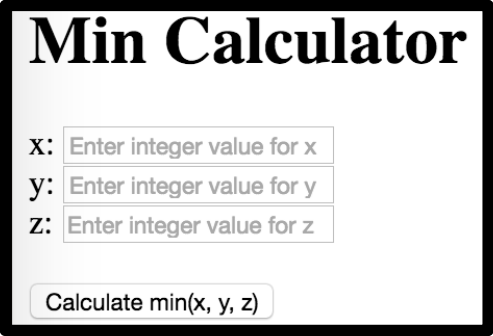
Enables automated testing of web apps

Has two key components:

- WebDriver
  - Browser-based regression tests
- IDE
  - Firefox add-on for record-and-replay

# A simple webpage to test

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta name="Course" content="UT EE360T (Spring
5     <title>Min Calculator</title>
6     <script>
7       function compute() {
8         var x = document.getElementById("x").value;
9         var y = document.getElementById("y").value;
10        var z = document.getElementById("z").value;
11        if (x == "" || isNaN(x) || y == "" || isNaN(y) || z == "" || isNaN(z)) {
12          document.getElementById("result").textContent = "Please enter integer values only!";
13          return;
14        }
15        var result = minimum(x, y, z);
16        document.getElementById("result").textContent =
17          "min(" + x + ", " + y + ", " + z + ") = " + result;
18      }
19      function minimum(x, y, z) {
20        return Math.min(Math.min(x, y), Math.min(x, z));
21      }
22    </script>
23  </head>
24  <body>
25    <h1>Min Calculator</h1>
26    x: <input id="x" placeholder="Enter integer value for x"><br>
27    y: <input id="y" placeholder="Enter integer value for y"><br>
28    z: <input id="z" placeholder="Enter integer value for z"><br><br>
29    <input type="button" id="computeButton" value="Calculate min(x, y, z)" onclick="compute()">
30    <br>
31    <h2 id="result"></h2>
32  </body>
33 </html>
```



# Selenium WebDriver: example



?/!