Course overview: the big picture

- **Software architecture and design**
  - Software modelling and UML crash course.
  - Best practices and OO design principles.
  - Architecture and Design patterns.

- **Empirical Software Engineering**
  - Reasoning about experimental designs and studies.
  - Understanding and reasoning about threats to validity.

- **Software testing and debugging**
  - Learning about cutting-edge research.
  - Hands-on experience, using testing and debugging techniques.

- **Class project**
  - Empirical study, development of a research prototype, etc.

---

Today

**Program analysis**

- What is program analysis?
- Why program analysis?
- Code review/inspection
- Static vs. dynamic analysis
<table>
<thead>
<tr>
<th>What is program analysis?</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>(Automatically) analyze the behavior of a program</em></td>
</tr>
<tr>
<td>○ Optimize the program or</td>
</tr>
<tr>
<td>○ check program’s behavior (against its specification)</td>
</tr>
<tr>
<td>• Concerned with properties such as</td>
</tr>
<tr>
<td>○ Correctness</td>
</tr>
<tr>
<td>○ Safety</td>
</tr>
<tr>
<td>○ Liveness</td>
</tr>
<tr>
<td>○ Performance</td>
</tr>
<tr>
<td>• Can be static or dynamic or a combination of both</td>
</tr>
</tbody>
</table>

What’s the difference between a static analysis and a dynamic analysis?

<table>
<thead>
<tr>
<th>Why do we need program analysis?</th>
</tr>
</thead>
<tbody>
<tr>
<td>![Image of airplane]</td>
</tr>
</tbody>
</table>
Why do we need program analysis?

- ~15 million lines of code
- Let's say 50 lines per page (0.05 mm)

Why do we need program analysis?

- ~15 million lines of code
- Let's say 50 lines per page (0.05 mm)
  - 300,000 pages
  - 15 m (49 ft)
Why do we need program analysis?

- Increase confidence in program correctness
- Understand the program’s behavior
- Prove properties about the program

Code review/inspection

Different types of reviews
- Code/design review
- Informal walkthrough
- Formal inspection

```java
double avg(double[] d) {
    int n = d.length;
    double s = 0;
    int i = 0;
    while (i<n)
        s = s + d[i];
        i = i + 1;
    double a = s / n;
    return a;
}
```

Anything that could be improved in this code?
### Code review/inspection

**Different types of reviews**
- Code/design review
- Informal walkthrough
- Formal inspection

```java
double avg(double[] nums) {
    int n = nums.length;
    double sum = 0;
    int i = 0;
    while (i < n)
        sum = sum + nums[i];
    i = i + 1;
    double avg = sum / n;
    return avg;
}
```

**Anything that could be improved in this code?**

```c
static OSStatus
SSLVerifySignedServerKeyExchange(...) {
    OSStatus err;
    ...
    if ((err = SSLHashSHA1.update(&hashCtx, &clientRandom)) != 0)
        goto fail;
    if ((err = SSLHashSHA1.update(&hashCtx, &serverRandom)) != 0)
        goto fail;
    if ((err = SSLHashSHA1.update(&hashCtx, &signedParams)) != 0)
        goto fail;
    goto fail;
    if ((err = SSLHashSHA1.final(&hashCtx, &hashOut)) != 0)
        goto fail;
    err = sslRawVerify(ctx, ctx->peerPubKey, dataToSign, dataToSignLen, signature, signatureLen);
    if (err) {
        SSLFreeBuffer(&signedHashes);
        SSLFreeBuffer(&hashCtx);
        return err;
    }
    fail:
        SSLFreeBuffer(&signedHashes);
        SSLFreeBuffer(&hashCtx);
        return err;
    }
```

**Anything wrong with that code?**

Apple's “goto fail” bug: A security vulnerability for 2 years!
Code review/inspection

Pros
- Can be applied at any step in the development process
- Improves confidence and communication

Cons
- Time-consuming
- Mostly informal
- Not repeatable

Static vs. dynamic analysis

Static analysis
- Build an abstraction of run-time states (and prove a property of the program)
- Reason about the program without executing it

Dynamic analysis
- Execute the program with some inputs
- Observe behavior

Static analysis: examples
- Type checking of a compiler
- Rule/pattern-based analysis (PMD, Findbugs, etc.)

```java
double avg(double[] nums) {
    int n = nums.length;
    double sum = 0;
    int i = 0;
    while (i<n)
        sum = sum + nums[i];
    i = i + 1;
    double avg = sum / n;
    return avg;
}
```

Static analysis: examples
- Type checking of a compiler
- Rule/pattern-based analysis (PMD, Findbugs, etc.)

```java
double avg(double[] nums) {
    int n = nums.length;
    double sum = 0;
    int i = 0;
    while (i<n)
        sum = sum + nums[i];
    i = i + 1;
    double avg = sum / n;
    return avg;
}
```
Static analysis: examples

- Control-flow analysis
- Data-flow analysis

double avg(double[] nums) {
  int n = nums.length;
  double sum = 0;
  int i = 0;
  while (i<n)
    sum = sum + nums[i];
    i = i + 1;
  double avg = sum / n;
  return avg;
}

What is the control flow graph (CFG) for this avg function?

Dynamic analysis: examples

A test for the avg function:

```
@Test
class TestAvg {
  @Test
  public void testAvg() {
    double[] nums = {1.0, 2.0, 3.0};
    double expected = 2.0;
    assertEquals(expected, avg(nums), EPS);
  }
}
```

Can we conclude that this is an infinite loop? Why?
<table>
<thead>
<tr>
<th>Static analysis vs. dynamic analysis</th>
<th>Program analysis</th>
</tr>
</thead>
<tbody>
<tr>
<td>See code examples (online)</td>
<td></td>
</tr>
<tr>
<td>● Compare the output of FindBugs and JUnit</td>
<td><strong>Static analysis</strong></td>
</tr>
<tr>
<td>○ What types of issues do these tools report?</td>
<td>● Build an abstraction of run-time states</td>
</tr>
<tr>
<td>○ Can these tools pinpoint the problem in the code?</td>
<td>(and prove a property of the program)</td>
</tr>
<tr>
<td>○ Does a reported error always indicate that something is wrong with the code?</td>
<td>● Reason about the program without execution</td>
</tr>
<tr>
<td>○ Does no reported error indicate that there is nothing wrong with the code?</td>
<td><strong>Dynamic analysis</strong></td>
</tr>
<tr>
<td></td>
<td>● Execute the program with some inputs</td>
</tr>
<tr>
<td></td>
<td>● Observe behavior</td>
</tr>
<tr>
<td></td>
<td>What about false positives and false negatives?</td>
</tr>
<tr>
<td></td>
<td><strong>Should we use static or dynamic analysis?</strong></td>
</tr>
</tbody>
</table>