**Scope and Memory Management**

CSCI 334  
Stephen Freund

---

**Type Inference Applications**
- Compilers  
  - are values used consistently with some type?  
- C++ template expansion  
  - must we generate a new template version?

- JVM Safety Checking
- Race condition analysis

---

**Programs on the Web**

Running Programs in a Browser

```
www.nasa.gov
```

- HTML
- applet
- files
- monitor
- printer
- network

---

```
<html>
  <applet>
    <!-- code here -->
  </applet>
</html>
```

Satellite Tracking Page

```
<html>
  <!DOCTYPE html>
  <head>
    <title>Satellite Tracking Page</title>
  </head>
  <body>
    <!-- code here -->
  </body>
</html>
```
Running Programs in a Browser

Sandbox Security Model

Enforcing Sandbox Boundaries

- **Problem**: Prevent direct access to resources
- **Enforcement** through type safety
  - permit library calls, but no "unsafe" operations
  - example:
    ```java
c   char *s = "moo";
c   s = s - 1000;  // BAD
   print s;
```
  - another example:
    ```java
c     byte b[] = { 0x12, 0xa3, 0x05, ... };
c     ((function)b());  // REALLY BAD
```

Using Type Safety for Security

- Compiler rejects programs with type errors:
- Why not sufficient for the Web?
Java vs. Java Bytecodes

class A extends Object {
    int i;
    void f(int val) { i = val + 1; }
}

Method void f(int)

0 aload 0
1 iload 1
2 iconst 1
3 iadd
4 putfield #4 <Field int i>
5 return

A obj.
0 100
Var 0

Stack

Var 1

Java vs. Java Bytecodes

class A extends Object {
    int i;
    void f(int val) { i = val + 1; }
}

Method void f(int)

0 aload 0
1 iload 1
2 iconst 1
3 iadd
4 putfield #4 <Field int i>
5 return

A obj.
0 100
Var 0

Var 1

Stack
Java vs. Java Bytecodes

class A extends Object {
    int i;
    void f(int val) { i = val + 1; }
}

Method void f(int)
0 aload 0
1 iload 1
2 iconst 1
3 iadd
4 putfield #4 <Field int i>
5 return

Java vs. Java Bytecodes

class A extends Object {
    int i;
    void f(int val) { i = val + 1; }
}

Method void f(int)
0 aload 0
1 iload 1
2 iconst 1
3 iadd
4 putfield #4 <Field int i>
5 return

Variable and Stack Types

Method void f(int)
0 aload 0
1 iload 1
2 iconst 1
3 iadd
4 putfield #4 <Field int i>
5 return

Does stack top have two integers?
### Stack and Type Variables

<table>
<thead>
<tr>
<th>Stack</th>
<th>Var 0 Type</th>
<th>Var 1 Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>S0</td>
<td>nil</td>
<td>V0 = A</td>
</tr>
<tr>
<td>S1</td>
<td>V0 = S0</td>
<td>W0 = int</td>
</tr>
<tr>
<td>S2</td>
<td>W1 = S1</td>
<td>V1 = V0</td>
</tr>
<tr>
<td>S3</td>
<td>int = S2</td>
<td>W2 = W1</td>
</tr>
<tr>
<td>S4</td>
<td>int = A</td>
<td>V4 = V3</td>
</tr>
<tr>
<td>S5</td>
<td>'b</td>
<td>W5 = W4</td>
</tr>
</tbody>
</table>

### Method void f(int)

```java
0 aload 0
1 iload 1
2 iconst 1
3 iadd
4 putfield #4 <Field int i>
5 return
```

### Processor Clock Speeds

![Processor Clock Speeds Chart](chart.png)

### Intel 4004 (1971)
- Intel 4004
- Intel 8086 (1978)

- Intel 4004
- Intel 8086
- Intel Pentium 4

- 2,300 transistors
- 50,000 transistors
- 50,000,000 transistors

### Intel Core i7 (2010)
- Intel Core i7
- 2,000,000,000 transistors
Multi-Core Chips

Concurrent Programming With Threads

Concurrent Programming With Threads

Multithreaded Program Execution

Multithreaded Program Execution
Race Condition

Thread A
...  
\[ t1 = bal; \]
\[ bal = t1 + 100; \]
...  
Thread B
...  
\[ t2 = bal; \]
\[ bal = t2 - 100; \]
...

\[ bal = 500 \]
\[ bal is 500 \]

Avoiding Race Conditions

Thread A
\[ acquire(m); \]
\[ t1 = bal; \]
\[ bal = t1 + 100; \]
\[ release(m); \]

Thread B
\[ acquire(m); \]
\[ t2 = bal; \]
\[ bal = t2 - 100; \]
\[ release(m); \]

• Common, Hard to Detect, Costly to Fix

Type Inference to Identify Races

Thread 1
\[ synchronized(l) \{ \]
\[ x := 10; \]
\[ \} \]
\[ synchronized(m) \{ \]
\[ synchronized(l) \{ \]
\[ x := !y + 1; \]
\[ \} \]
\[ y := 2; \]
\[ \} \]

Thread 2
\[ synchronized(m) \{ \]
\[ print !x; \]
\[ \} \]
\[ synchronized(m) \{ \]
\[ synchronized(l) \{ \]
\[ x := !y + 1; \]
\[ \} \]
\[ y := 2; \]
\[ \} \]