Programming with Threads

Benefits and Risks

How can we perform several computations concurrently?
Benefits

- Model concurrent tasks
- Do this but also that
- Exploit multiple processors
- Shorten execution time
- Handle asynchronous events
- When waiting, do something else
Risks

- Thread-safety
  - Nothing bad should happen
- Liveness
  - Something good should happen
- Performance
  - It should happen quickly
Risks

Thread-safety — Description

”A class is thread-safe if it behaves correctly when accessed from multiple threads, [...] with no additional synchronization [...] on the part of the calling code.”
Risks

### Thread-safety – Example

```java
public class UnsafeSequence {
    private int value;

    public int getNext() {
        return value++;
    }
}
```

The diagram illustrates the sequence of operations: load, add, and store. There are two threads, A and B, that access the same variable `value`. If thread A starts with `value` being 5 and thread B also reads the same value, both threads will increment it to 6, resulting in a race condition where the final value of `value` is not deterministic. This highlights the importance of thread safety in concurrent programming.
Thread-safety — Solution

Three ways to fix safety

- Don’t share state — need counter
- Make state immutable — need increment
- Use synchronization — okay
Risks

Thread-safety — Example

```java
public class Sequence {
    private int value;

    public synchronized int getNext() {
        return value++;
    }
}
```

### Diagram

```
A: L | value → 5 | 5+1 → 6 | value = 6 | U

B: L | value → 6 | 6+1 → 7 | value = 7 | U
```
Risks

Liveness — Description

”A liveness failure occurs when an activity gets into a state such that it is permanently unable to make forward progress.”
Risks

Liveness – Example

Deadlock
Not only does synchronization make selected parts of the execution sequential, it also adds overhead when acquiring and releasing the locks.
Threads – Summary

Benefits
- Model concurrent tasks
- Exploit multiple processors
- Handle asynchronous events

Risks
- Safety
- Liveness
- Performance
This slide is intentionally left blank-ish.