



Threads (Practice)

**ICS332
Operating Systems**

Henri Casanova (henric@hawaii.edu)



(q1) Basic True/False Questions

- Two threads created by the same process share the same stack
- Two threads created by two different processes share the same heap
- Two threads created by the same process can share data on the heap
- If a CPU-bound thread runs in 1h on a single core, then two copies of that same thread running together on that same single core will run in more than 1h
- True concurrency is necessary for interactivity
- If all our CPU instantly became single-core, we would no longer use threads

(q1) Basic True/False Questions

- Two threads created by the same process share the same stack [FALSE]
- Two threads created by two different processes share the same heap [FALSE]
- Two threads created by the same process can share data on the heap [TRUE]
- If a CPU-bound thread runs in 1h on a single core, then two copies of that same thread running together on that same single core will run in more than 1h [TRUE]
- True concurrency is necessary for interactivity [FALSE]
- If all our CPU instantly became single-core, we would no longer use threads [FALSE]

(q2) What does this print?

```
public static void main() {  
    Thread t1 = new Thread(() -> { // in-line thread creation  
        for (int i = 1; i <= 10; i++) { System.out.print("X"); }  
    });  
  
    Thread t2 = new Thread(() -> { // in-line thread creation  
        for (int i = 1; i <= 5; i++) { System.out.print("Y"); }  
        try { t1.join(); } catch (InterruptedException &ignore) {}  
        for (int i = 1; i <= 5; i++) { System.out.print("Y"); }  
    });  
    a.start(); b.start();  
}
```

- Describe the structure of the output (there are ~3,000 possible output strings)

(q2) Answer

```
public static void main() {  
    Thread t1 = new Thread(() -> { // in-line thread creation  
        for (int i = 1; i <= 10; i++) { System.out.print("X"); }  
    });  
  
    Thread t2 = new Thread(() -> { // in-line thread creation  
        for (int i = 1; i <= 5; i++) { System.out.print("Y"); }  
        try { t1.join(); } catch (InterruptedException &ignore) {}  
        for (int i = 1; i <= 5; i++) { System.out.print("Y"); }  
    });  
    a.start(); b.start();  
}
```

- A string of 10 X's and 5 Y's in any order, followed by 5 Y's