Monitors
(1 February)

Which of the following are possible signaling disciplines in monitors?

1. Signal and wait
2. Signal and urgent wait
3. Signal and continue
4. Urgent signal and continue

Which of the following are possible signaling disciplines in monitors?

1. Signal and wait
2. Signal and urgent wait
3. Signal and continue
4. Urgent signal and continue

What does a call to method "`print`" print?

```
monitor class CountPrint {
  private int count = 0;
  private Condition isTwo = new Condition();
  public void inc() {
    count += 1;
    if (count == 2) isTwo.signal();
  }
  public void print() {
    if (count != 2) isTwo.wait();
    System.out.println(count);
  }
}
```

1. It always prints `"2"`.
2. It prints `"2"` if the monitor uses "signal and wait".
3. It prints `"1"` or `"2"`.
4. If the monitor uses "signal and continue" it may print `"3"`.

What does a call to method “`print`” print?

```
monitor class CountPrint {
  private int count = 0;
  private Condition isTwo = new Condition();
  public void inc() {
    count += 1;
    if (count == 2) isTwo.signal();
  }
  public void print() {
    if (count != 2) isTwo.wait();
    System.out.println(count);
  }
}
```

1. It always prints `"2"`.
2. It prints `"2"` if the monitor uses “signal and wait”.
3. It prints `"1"` or `"2"`.
4. If the monitor uses “signal and continue” it may print `"3"`.

What does a call to method "`print`" print?

```
monitor class CountPrint {
  private int count = 0;
  private Condition isTwo = new Condition();
  public void inc() {
    count += 1;
    if (count == 2) isTwo.signal();
  }
  public void print() {
    while (count != 2) isTwo.wait();
    System.out.println(count);
  }
}
```

1. It always prints `"2"`.
2. It prints `"2"` or it may block forever.
3. It prints `"1"` or `"2"`.
4. If the monitor uses "signal and continue" it may print `"3"`.

What does a call to method "`print`" print?

```
monitor class CountPrint {
  private int count = 0;
  private Condition isTwo = new Condition();
  public void inc() {
    count += 1;
    if (count == 2) isTwo.signal();
  }
  public void print() {
    while (count != 2) isTwo.wait();
    System.out.println(count);
  }
}
```

1. It always prints `"2"`.
2. It prints `"2"` or it may block forever.
3. It prints `"1"` or `"2"`.
4. If the monitor uses "signal and continue" it may print `"3"`.

What does the following two-thread program print?

```
Barrier barrier = new Barrier(2); // barrier for two threads
---------------------------------------------------------------------
          thread t                 |            thread u
System.out.prinln("t");            | barrier.wait();
barrier.wait();                    | System.out.prinln("u");
```

1. It prints `"t"` followed by `"u"`.
2. It prints `"u"` followed by `"t"`.
3. Either of the answers above.
4. Either of the answers above or it does not print anything.

What does the following two-thread program print?

```
   Barrier barrier = new Barrier(2); // barrier for two threads
   ------------------------------------------------------------
              thread t                      |           thread u
1  System.out.prinln("t");                  | barrier.wait();                3
2  barrier.wait();                          | System.out.prinln("u");        4
```

1. It prints `"t"` followed by `"u"`.
2. It prints `"u"` followed by `"t"`.
3. Either of the answers above.
4. Either of the answers above or it does not print anything.

What does the following two-thread program print?

```java
Barrier barrier = new Barrier(2); // barrier for two threads
```

	thread `t`	thread `u`	
1	`barrier.wait();`	`barrier.wait();`	3
2	`System.out.prinln("t");`	`System.out.prinln("u");`	4

1. It prints `"t"` followed by `"u"`.
2. It prints `"u"` followed by `"t"`.
3. Either of the answers above.
4. Either of the answers above or it does not print anything.

What does the following two-thread program print?

```
Barrier barrier = new Barrier(2); // barrier for two threads
-----------------------------------------------------------------------
            thread t                 |          thread u
                                     |
1  barrier.wait();                   |  barrier.wait();                3
2  System.out.prinln("t");           |  System.out.prinln("u");        4
```

1. It prints `"t"` followed by `"u"`.
2. It prints `"u"` followed by `"t"`.
3. Either of the answers above.
4. Either of the answers above or it does not print anything.