

Q: What's wrong with the following code, and how would you fix it?

```
public void fillByteArray(InputStream s, byte[] b, int offset,
                          Integer numReads, int readAmount){
    while (numReads > 0){
        s.read(b, offset, readAmount);
        offset += readAmount;
        numReads--;
    }
}
```

Q: Write a piece of code that starts 2 threads that are **guaranteed** to deadlock on every execution of the code.

Q: Java string are immutable, what advantages does this give you as a programmer? (name as many as you can)

Q: What's the difference between the following:

- long x;
- volatile long y;
- AtomicLong z;

and when would you use each one?

Q: What's the best way to sleep/park a thread and then wake it up on demand? (Using pseudo-code is ok)

Q: A veterinary practice wants to keep track of its patients and their humans. Consider the following database table definition. What problems might this design cause, and how would you change the design to prevent these problems?

```
CREATE TABLE patient (  
    patient_name      VARCHAR(255),  
    patient_dob       DATETIME,  
    patient_species   VARCHAR(255),  
    patient_breed     VARCHAR(255),  
    patient_gender    VARCHAR(255),  
    customer_first_name VARCHAR(255),  
    customer_last_name VARCHAR(255),  
    customer_address  VARCHAR(255),  
    customer_phone    VARCHAR(255),  
    customer_credit_card VARCHAR(255),  
    customer_cc_expiry DATE  
);
```

Q: When / where would you use the following:

- -Dname=value
- Integer.getInteger("name")
- System.getenv("name")
- System.getProperty("name")
- new InitialContext();
- @Resource
- this.getClass().getResourceAsStream("foo")
- Thread.currentThread().getContextClassLoader().getResourceAsStream("foo")

Q: What does the Java expression (Integer.MIN_VALUE * -1) evaluate to, and why?

Q: What are some reasons you might want to write an interface for a class, even if you don't need more than one implementation?

Q: What can go wrong if you override hashCode() but not equals(), or vice versa?