Sample Midterm Exam

CS474 Object-Oriented Languages and Environments

Spring 2023

Name:	UIN:

Instructions: Fill in your name and UIN in the cover page and on every page. Read each question carefully. Please keep your answer within the provided boxes. This exam has a total of 6 questions (16 parts) worth 100 points.

Question 1. (25 points) Consider the following Java code:

```
class InputStream {
    byte readByte() throws IOException { ... }

    void close() { ... }
}

class ZeroInputStream extends InputStream {
    byte readByte() throws IOException { return 0; }

    void close() { /* empty */ }
}

InputStream a = ... ?
```

1.1. (10 points) Just by calling methods `readByte` and `close`, can you tell the **dynamic type** of variable `a`? Explain your answer.

1.2. (5 points) Is method `readByte` overloaded or overridden? How do you know?

1.3. (10 points) The following code attempts to copy the contents of an `InputStream` to an array of bytes, and returns the number of bytes written. Modify the code to ensure that:

1. The array is never accessed out of bounds.
2. The `InputStream` is always closed.
3. The code handles all checked exceptions.

```
int copy(InputStream in, byte[] b) {
  int i = 0;
  while (true) { b[i++] = in.read(); }
  return i;
}
```

Question 2. (30 points) Consider the following Java code:

```
class WritingCodeLikeThisMakesMeSad {
  public static int sum(List l) {
    int result = 0;
    List iter = l;
    while (iter != null) {
      result += iter.item;
      iter = iter.next;
    }
    return result;
}
```

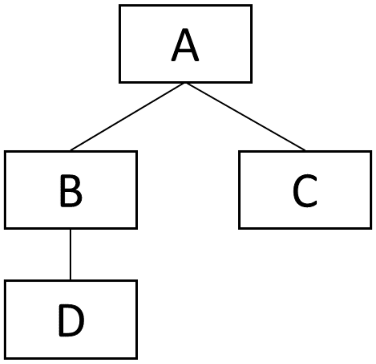
```
class List {
  private int item;
  private List next;

  public Iterator iter()
    { ... }
}

interface Iterator {
  boolean hasNext();
  int next();
}
```

2.1. (10 points) Name the **fundamental principle** of Object-Oriented design that method *sum* violates.

2.2. (10 points) List one **disadvantage** of violating the principle stated above.

2.3. (10 points) **Rewrite** the code above to not violate that principle.

Question 3. (10 points) Consider the following class hierarchy:



3.1. (5 points) Give an example of an upcast, a downcast, and an invalid (sideways) cast in this hierarchy.

3.2. (5 points) Will the following code succeed, fail at compile time, or fail at runtime? Explain why.

```
A a = new D();
C c = (C) a;
```

Question 4. (15 points) Executing the following Java code:

```
class C { public void m() { } }

for (java.lang.reflect.Method m : C.class.getMethods()) {
    System.out.print(m.getName() + ",");
}
```

Results in the following output:

```
m, wait, equals, toString, hashCode, getClass, notify, notifyAll,
```

4.1. (15 points) **Explain** the behavior described above.

Question 5. (10 points) Suppose we have `int i = 4, j = 4`. For each of the following comparisons, does it return `true` or `false`?

5.1. (2 points) `i == j`

5.2. (2 points) `new Integer(i) == new Integer(j)`

5.3. (2 points) `new Integer(i).equals(new Integer(4))`

5.4. (2 points) `new Integer(i).equals(new Integer(j))`

5.5. (2 points)

```
Integer k = new Integer(i);
Integer m = k;
k == m
```

Question 6. (10 points) Consider the following code:

```
interface Printable {
  default void write(String s) { System.out.println(s); }
}

interface Storable {
  static String value;
  default void write(String s) { value = s; }
}

class Data implements Storable, Printable { }

Data d = new Data();
d.write("hi!");
```

6.1. (5 points) Will this code succeed, fail at compile time, or fail at runtime? Explain why.

6.2. (5 points) How would we need to change the code so that `Data` objects use the `write` method from `Printable`?