This quiz (7 pages) is designed to test your knowledge of basic, but tricky Java features. Don’t get worried about how you do on the quiz—this is for fun, and to get you thinking about Java again. However, if you do well, the quiz score can benefit your final grade.

We will use the results of the quiz to understand how well everyone understands Java in the class. Thursday’s lecture will cover material that appears on this quiz.

<table>
<thead>
<tr>
<th>Question</th>
<th>Points</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>25</td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>100</td>
<td></td>
</tr>
</tbody>
</table>
1. (Objects, 15 points)

```java
class MyInt {
    int x;
    public MyInt(int x) { this.x = x; }
}
class Foo {
    void inc(int x) { x = x+1; }
    void inc(MyInt o) { o.x = o.x+1; }
}
```

Given the above class definitions, what is the output of the following code sequence:

```java
MyInt o = new MyInt(5);
int x = 5;
Foo f = new Foo();

f.inc(x);
f.inc(o);
System.out.println(x);
System.out.println(o.x);
```

**Answer:**

```
5
6
```
Given the three classes shown above, write the output of executing the following code sequence.

```java
class Foo1 {
    private int x;
    Foo1(int x) { this.x = x; }
}
class Foo2 {
    private int x;
    Foo2(int x) { this.x = x; }
    boolean equals(Foo2 o) {
        return (this.x == o.x);
    }
}
class Foo3 {
    private int x;
    Foo3(int x) { this.x = x; }
    boolean equals(Object o) {
        if (o instanceof Foo3) {
            return (this.x == ((Foo3)o).x);
        } else return false;
    }
}

Foo1 f1a = new Foo1(1);
Foo1 f1b = new Foo1(1);
Object o = f1a;
System.out.println(f1a == f1a);
System.out.println(f1a == o);
System.out.println(f1b == o);
System.out.println(f1a.equals(f1a));
System.out.println(f1a.equals(o));
System.out.println(f1b.equals(f1a));

Foo2 f2a = new Foo2(1);
Foo2 f2b = new Foo2(1);
Object o = f2a;
System.out.println(f2a.equals(o));
System.out.println(f2b.equals(o));
System.out.println(o.equals(f2b));
System.out.println(f2b.equals(f2a));

Foo3 f3a = new Foo3(1);
Foo3 f3b = new Foo3(1);
Object o = f3a;
System.out.println(f3a.equals(f3a));
System.out.println(f3a.equals(o));
System.out.println(o.equals(f3a));
System.out.println(f3b.equals(f3a));

Answer:
```
true
true
false
true
false
true
false
false
true
true
true
true
true
true
true
3. (Subtyping, 30 points)

Given the above class and interface relationships, what will be the output of the following code sequence? Recall that `o instanceof C` returns `true` when the actual run-time type of `o` is a subtype of `C`. If any of the `instanceof` checks fail to compile, indicate which ones.

```java
Foo f = new Foo();
System.out.println(f instanceof Foo);
System.out.println(f instanceof FooIfc);

Bar b = new Bar();
System.out.println(b instanceof FooIfc);
System.out.println(b instanceof BarIfc);
System.out.println(b instanceof Object);

Object o = b;
System.out.println(o instanceof Foo);
System.out.println(o instanceof Bar);

FooIfc f2 = new Bar();
System.out.println(f2 instanceof Foo);
System.out.println(f2 instanceof FooIfc);

Bif b2 = new Bif();
System.out.println(b2 instanceof FooIfc);
System.out.println(b2 instanceof Bar);
System.out.println(b2 instanceof BarIfc);
System.out.println(b2 instanceof BifIfc);
```

**Answer:**

The statement `System.out.println(b2 instanceof Bar)` does not compile; if removed the output would be:

```
true
true
true
true
true
true
true
true
true
true
```
true
false
true
4. (Exceptions, 25 points)

class AppException extends Exception {}  
class AppFault extends AppException {}  
class AppError extends Error {}  

(a) Which of the following methods will cause errors during compilation:

class Foo {
  void f1() { throw new Exception(); }
  void f2() { throw new AppException(); }
  void f3() { throw new AppError(); }
  void f4() throws AppException { throw new Exception(); }
  void f5() throws AppException { throw new AppFault(); }
}

Answer:

Methods f1, f2, and f4 fail: they do not declare thrown exceptions.

(b) What will be the output of executing the main method of the Foo class defined next:

class Foo {
  void f1() throws Exception { throw new AppFault(); }
  void f2() { throw new AppError(); }

  public static void main(String args[]) {
    Foo f = new Foo();
    try { f.f1() }
    catch (AppFault e) {
      System.out.println("Got AppFault");
    }
    catch (Exception e) {
      System.out.println("Exception");
    }
    finally {
      System.out.println("Done");
    }
    try { f.f2() }
    catch (Exception e) {
      System.out.println("Exception");
    }
    finally {
      System.out.println("Done");
    }
  }
}

Answer:

Got AppFault
Done
Done

program then exits due to an uncaught AppError