The 5th toughest question.

26. Given this code:

```java
public class Confused {
    private int n1, n2, n3;

    public Confused(int n1, int n2, int n3) {
        n2 = n1;
        this.n2 = n3;
        this.n3 = n2;
        n3 = n1;
    }

    public Confused(int n1, int n2) {
        this(n2, n1, n1);
    }

    public String toString() {
        return n1 + " " + n2 + " " + n3;
    }

    public static void main(String[] args) {
        Confused c1 = new Confused(5, 6, 7);
        System.out.println("c1 = " + c1);

        Confused c2 = new Confused(8, 9);
        System.out.println("c2 = " + c2);
    }
}
```

What is the second line of output?

A. `c2 = 0 8 9`
B. `c2 = 0 9 8`
C. `c2 = 8 8 9`
D. `c2 = 9 9 8`
E. `c2 = 9 9 9`

% answered (A): 38.5% (Correct answer)
% answered (B): 14.8%
% answered (C): 14.1%
% answered (D): 15.6%
% answered (E): 14.1%
% unanswered: 3.0%
**Explanation:**

When this statement is executed:

```
Confused c2 = new Confused(8, 9);
```

An object `c2` is instantiated, with its data members initialized to zero:

<table>
<thead>
<tr>
<th>n1</th>
<th>n2</th>
<th>n3</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

The constructor `Confused(8, 9)` is called, which in turn calls `Confused(9, 8, 8)`. Hence the parameters `n1`, `n2` and `n3` take the following values:

<table>
<thead>
<tr>
<th>n1</th>
<th>n2</th>
<th>n3</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>8</td>
<td>8</td>
</tr>
</tbody>
</table>

The statement `n2 = n1;` results in this:

```
Confused (int n1, int n2, int 3)
```

<table>
<thead>
<tr>
<th>n1</th>
<th>n2</th>
<th>n3</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>9</td>
<td>8</td>
</tr>
</tbody>
</table>

The statement `this.n2 = n3;` results in this:

```
Confused (int n1, int n2, int 3)
```

<table>
<thead>
<tr>
<th>n1</th>
<th>n2</th>
<th>n3</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>8</td>
<td>8</td>
</tr>
</tbody>
</table>
The statement `this.n3 = n2;` results in this:

![Diagram showing the state of `n1`, `n2`, and `n3` after the statement `this.n3 = n2;`]

The statement `n3 = n1;` results in this:

![Diagram showing the state of `n1`, `n2`, and `n3` after the statement `n3 = n1;`]

Hence when the method is done, the object `c2` contains 0, 8, 9 in its data members `n1`, `n2`, `n3` respectively.
The 4th toughest question.

9. Given the following code, which of the following statements is true?

```java
public class Circle {
    private double radius;

    public Circle (double radius) {
        radius = radius;
    }

    public static void main (String[] args) {
        . . .
    }
}
```

A. The program can compile, but you cannot create a Circle object with a specified radius.
B. The program can compile, but you cannot create a Circle object by using the statement
   ```java
   Circle c = new Circle(5.2);
   ```
   which will give a run-time error.
C. The program cannot compile because the constructor should not contain any formal parameter.
D. The program cannot compile because of the error in the assignment statement.
E. The program cannot compile because there is no accessor or mutator method.

% answered (A): 32.6% (Correct answer)
% answered (B): 11.1%
% answered (C): 6.7%
% answered (D): 42.2%
% answered (E): 6.7%
% unanswered: 0.7%

Explanation:
There is no error (compilation or run-time) with this code. Since the assignment statement:

```java
radius = radius;
```

merely accesses the parameter instead of the data member, the answer is (A). The correct statement should be:

```java
this.radius = radius;
```
The 3rd toughest question.

5. What is the output of the following code fragment?

```java
int a, b = 7, c = 5;
if (b++ > 7) a = ++b + c++;
else         a = ++b – c++;
System.out.println("" + a + " " + b + " " + c);
```

<table>
<thead>
<tr>
<th>Option</th>
<th>Output</th>
<th>% Answered</th>
</tr>
</thead>
<tbody>
<tr>
<td>A. 14 9 6</td>
<td>% Answered (A): 17.8%</td>
<td></td>
</tr>
<tr>
<td>B. 14 8 6</td>
<td>% Answered (B): 17.8%</td>
<td></td>
</tr>
<tr>
<td>C. 4 9 6</td>
<td>% Answered (C): 31.9% (Correct Answer)</td>
<td></td>
</tr>
<tr>
<td>D. 3 9 6</td>
<td>% Answered (D): 19.3%</td>
<td></td>
</tr>
<tr>
<td>E. 2 8 6</td>
<td>% Answered (E): 10.4%</td>
<td></td>
</tr>
</tbody>
</table>

% Unanswered: 3.0%

Explanation:
The expression `(b++ > 7)` is false, since the current value of `b` is used in the comparison. Since the current value of `b` is 7, which is not greater than 7, the expression is false. The value of `b` is then incremented to 8.

The ‘else’ part of the ‘if’ statement is executed. Since `b` is already 8 (from above), `++b` gives 9. Since we use the post-increment operator `c++`, the current value of `c` (which is 5) is used in the subtraction before we increment `c`. Hence `a = 9 – 5`, or 4. The variable `c` is incremented to 6 after this.
The 2nd toughest question.
21. Given the following codes, what is the output?

```java
public class Guess {
    private static int value1 = 123;
    private int value2;

    public void increment() {
        value1++; value2++;
    }

    public int getValue1() {
        return value1;
    }

    public int getValue2() {
        return value2;
    }
}

public class DemoGuess {
    public static void main(String[] args) {
        Guess guess1 = new Guess();
        Guess guess2 = new Guess();
        guess1.increment();
        guess2.increment();
        System.out.println(guess1.getValue1() + " " +
                          guess1.getValue2() + " " +
                          guess2.getValue1() + " " +
                          guess2.getValue2());
    }
}
```

<table>
<thead>
<tr>
<th>Answer</th>
<th>Percentage Answered</th>
</tr>
</thead>
<tbody>
<tr>
<td>A. 124 1 125 2</td>
<td>43.0%</td>
</tr>
<tr>
<td>B. 125 1 125 1</td>
<td>29.6% (Correct answer)</td>
</tr>
<tr>
<td>C. 124 2 125 2</td>
<td>13.3%</td>
</tr>
<tr>
<td>D. 125 2 125 2</td>
<td>10.4%</td>
</tr>
<tr>
<td>E. 125 125 125 125</td>
<td>3.0%</td>
</tr>
<tr>
<td><strong>Unanswered</strong></td>
<td>0.7%</td>
</tr>
</tbody>
</table>

Explanation:
Note that there is a class data member `value1` (which is initialized to 123), and an instance data member `value2` that is attached to each object of the class `Guess`. The following is the result after the two statements:

```java
Guess guess1 = new Guess();
Guess guess2 = new Guess();
```
The statement `guess1.increment();` results in this:

```
Guess
value1 124
guess1 : Guess
value2 1
guess2 : Guess
value2 0
```

The statement `guess2.increment();` results in this:

```
Guess
value1 125
guess1 : Guess
value2 1
guess2 : Guess
value2 1
```
The toughest question.

30. What is the output of the following code?

```java
Fraction f1 = new Fraction(1, 2);
Fraction f2 = new Fraction(2, 3);
Fraction f3;

f1.addVersion2(f2.addVersion2(f1));
System.out.println(f1);
```

A. 20 12  
B. 20/12  
C. 5/3  
D. 1.666666666666667  
E. Error

% answered (A): 3.7%  
% answered (B): 42.2%  
% answered (C): 14.1%  
% answered (D): 5.9%  
% answered (E): 28.1% (Correct answer)  
% unanswered: 5.9%

Explanation:
The answer is rather straightforward. The `addVersion2` method is a void method, therefore `f2.addVersion2(f1)` – let’s call it X – does not return any value, and hence `f1.addVersion2(X)` is illegal.