Explanation A

The method *getOffset(hoursOffset, minsOffset)* calculates the offset in milliseconds given the hours and minutes offset. The assertion in line 4 checks if the method outputs the expected value of −8100000.
Internally, the method computes the value of the variable *offset* based on its *hoursOffset* and *minsOffset* parameters.

However,
1. In V1, line 10, the method reassigns *minsOffset* to the value of *hoursOffset*60 – Math.abs(minsOffset)*.
2. In V2, due to the change in line 10, it reassigns *minsOffset* to the value of *hoursOffset*60 – minsOffset,* omitting the *abs* operation.

Then, in both versions, it calculates the value of *offset* as
a function of the *minsOffset* value.

In summary, the change in the initial value assigned to *minsOffset* in line 10 (i.e., the removal of the call to the *Math.abs* method for the value of *minsOffset*), leads to a difference in computing the final value of o*ffset* in V1 and V2.

This difference causes the *offset* to differ from − 8100000 in line 4.

Explanation B

The method *getOffset(hoursOffset, minsOffset)* calculates the offset in milliseconds given the hours and minutes offset. The assertion in line 4 checks if the method outputs the expected value of −8100000.
Internally, the method computes the value of the variable *offset* based on its *hoursOffset* and *minsOffset* parameters,
which are initialized in lines 2 and 3 to -2 and -15, respectively.
However,
1. In V1, line 10, the method reassigns *minsOffset* to the value of *hoursOffset*60 – Math.abs(minsOffset)*.
2. In V2, due to the change in line 10, it reassigns *minsOffset* to the value of *hoursOffset*60 – minsOffset,* omitting the *abs* operation.

Then, in both versions, it calculates the value of *offset* as
a function of the *minsOffset* value.

In summary, the change in the value assigned to *minsOffset* in line 10 (i.e., the removal of the call to the *Math.abs* method for the value of *minsOffset*), leads to a difference in computing the final value of o*ffset* in V1 and V2
when the value of *minsOffset* is -15.
This difference causes the function output to differ from − 8100000 in line 4.

Explanation C

The method *getOffset(hoursOffset, minsOffset)* calculates the offset in milliseconds given the hours and minutes offset. The assertion in line 4 checks if the method outputs the expected value of −8100000.
Internally, the method computes the value of the variable *offset* based on its *hoursOffset* and *minsOffset* parameters.

However,
1. In V1, line 10, the method reassigns *minsOffset* to the value of *hoursOffset*60 – Math.abs(minsOffset)*.
2. In V2, due to the change in line 10, it reassigns *minsOffset* to the value of *hoursOffset*60 – minsOffset,* omitting the *abs* operation.

Then, in both versions, it calculates the value of *offset* as
a multiplication of its initial value by the value of *secondsPerMin*, followed by the multiplication of the obtained value by the value of *millisPerSecond* (lines 13-17) .
In summary, the change in the initial value assigned to *minsOffset* in line 10 (i.e., the removal of the call to the *Math.abs* method for the value of *minsOffset*), leads to a difference in computing the final value of o*ffset* in V1 and V2.

This difference causes the *offset* to differ from − 8100000 in line 4.

Notations: Colored backgrounds highlight the differences between the views.

FYI: Views are given below again, for your reference.

View A

V1

```
1  public static void main(String[] args){
2
3
4  ✓ assert(-8100000, getOffset(hoursOffset, minsOffset));
5  }
6  public int getOffset(int hoursOffset, int minsOffset){
7
8
9
10     minsOffset = hoursOffset*60 – Math.abs(minsOffset);
11
12
13
14
15
16
17   offset = Func1(minsOffset);
18   return offset;
19 }
```

Update

V2

```
1  public static void main(String[] args){
2
3
4  ✗ assert(-8100000, getOffset(hoursOffset, minsOffset));
5  }
6  public int getOffset(int hoursOffset, int minsOffset){
7
8
9
10     minsOffset = hoursOffset*60 – minsOffset;
11
12
13
14
15
16
17   offset = Func1(minsOffset);
18   return offset;
19 }
```

View B

V1

```
1  public static void main(String[] args){
2    int hoursOffset = -2;
3    int minsOffset = -15;
4  ✓ assert(-8100000, getOffset(hoursOffset, minsOffset));
5  }
6  public int getOffset(int hoursOffset, int minsOffset){
7
8
9
10     minsOffset = hoursOffset*60 – Math.abs(minsOffset);
11
12
13
14
15
16
17   offset = Func1(minsOffset);
18   return offset;
19 }
```

Update

V2

```
1  public static void main(String[] args){
2    int hoursOffset = -2;
3    int minsOffset = -15;
4  ✗ assert(-8100000, getOffset(hoursOffset, minsOffset));
5  }
6  public int getOffset(int hoursOffset, int minsOffset){
7
8
9
10     minsOffset = hoursOffset*60 – minsOffset;
11
12
13
14
15
16
17   offset = Func1(minsOffset);
18   return offset;
19 }
```

View C

V1

```
1  public static void main(String[] args){
2
3
4  ✓ assert(-8100000, getOffset(hoursOffset, minsOffset));
5  }
6  public int getOffset(int hoursOffset, int minsOffset){
7
8
9
10     minsOffset = hoursOffset*60 – Math.abs(minsOffset);
11
12
13   int offset = minsOffset;
14
15   offset  = offset * secondsPerMin;
16
17   offset = offset * millisPerSecond;
18   return offset;
19 }
```

Update

V2

```
1  public static void main(String[] args){
2
3
4  ✗ assert(-8100000, getOffset(hoursOffset, minsOffset));
5  }
6  public int getOffset(int hoursOffset, int minsOffset){
7
8
9
10     minsOffset = hoursOffset*60 – minsOffset;
11
12
13   int offset = minsOffset;
14
15   offset  = offset * secondsPerMin;
16
17   offset = offset * millisPerSecond;
18   return offset;
19 }
```