Week 10 Tutorial

Brae

May 9, 2018

CSSE2002: Programming in the Large

Question One

```
public void f(int arr[]) {
    int total=0;
    for (int i=0;i<arr.length;++i) {
        total+=arr[i];
        i++;
    }
    System.out.println(total);
}
```

What would be output to the terminal if f was called with the array {1,2,3,3,5,6,7}?

Question One

```
public void f(int arr[]) {
    int total=0;
    for (int i=0;i<arr.length;++i) {
        total+=arr[i];
        i++;
    }
    System.out.println(total);
}
```

What would be output to the terminal if f was called with the array {1,2,3,3,5,6,7}?

Note: Terminal means screen/console/output window

Question One

```
public void f(int arr[]) {
    int total=0;
    for (int i=0;i<arr.length;++i) {
        total+=arr[i];
        i++;
    }
    System.out.println(total);
}
```

What would be output to the terminal if f was called with the array {1,2,3,3,5,6,7}?

Note that **i** is incremented in both the afterthought and loop body.

$1 + 3 + 5 + 7 = 16$

Question Two

```
public int g(int v) {
    if (v>5) {
        return 0;
    }
    if (v<=0) {
        return Math.abs(v)+g(v+2);
    }
    return (v-1)+g(v+1);
}
```

What would be returned by the following calls? g(2), g(0), g(-5)

Question Two

```
public int g(int v) {
    if (v>5) {
        return 0;
    }
    if (v<=0) {
        return Math.abs(v)+g(v+2);
    }
    return (v-1)+g(v+1);
}
```

What would be returned by the following calls? g(2), g(0), g(-5)

Recursive Desk Check - Can you do it?

Question Two

```
public int g(int v) {
    if (v>5) {
        return 0;
    }
    if (v<=0) {
        return Math.abs(v)+g(v+2);
    }
    return (v-1)+g(v+1);
}
```

What would be returned by the following calls? g(2), g(0), g(-5)

Recursive Desk Check - Can you do it?

Hint: Sometimes you can use your other working to come to a solution quicker

Question Two Answers

g(v)	returned	evaluated returned
g(2)	1 + g(3)	$1 + 9 = 10$
g(3)	2 + g(4)	$2 + 7 = 9$
g(4)	3 + g(5)	$3 + 4 = 7$
g(5)	4 + g(6)	$4 + 0 = 4$
g(6)	0	0

Question Two Answers

g(v)	returned	evaluated returned
g(2)	1 + g(3)	1 + 9 = 10
g(3)	2 + g(4)	2 + 7 = 9
g(4)	3 + g(5)	3 + 4 = 7
g(5)	4 + g(6)	4 + 0 = 4
g(6)	0	0
g(0)	0 + g(2)	0 + 10 = 10

Question Two Answers

g(v)	returned	evaluated returned
g(2)	1 + g(3)	1 + 9 = 10
g(3)	2 + g(4)	2 + 7 = 9
g(4)	3 + g(5)	3 + 4 = 7
g(5)	4 + g(6)	4 + 0 = 4
g(6)	0	0
g(0)	0 + g(2)	0 + 10 = 10
g(-5)	5 + g(-3)	5 + 14 = 19
g(-3)	3 + g(-1)	3 + 11 = 14
g(-1)	1 + g(1)	1 + 10 = 11
g(1)	0 + g(2)	0 + 10 = 10

Question Three

```
public class V {
    ...
    public static double f(int v);
    ...
}
```

Write a single JUnit4 test method which checks that:

- f(2) returns a value greater than 0.
- f(0) returns 0.5
- f(-1) throws a NullPointerException

Question Three

Exam Tips

- You can always assume any common imports have been imported.
- If you are asked to write code fragments, you can assume the class and containing methods are defined.
- If you are asked to write a method, write a whole method.
- If you are asked to write a program, write the whole class including a main method.
- If you believe that you are using types which are ambiguous, put a comment.

Question Three

```
@Test
public void test() {
    Assert.assertTrue(V.f(2)>0); // remember f is
        static
    Assert.assertEquals(0.5, V.f(0), 0.001);
    try {
        f(-1);
        Assert.fail();
    } catch (NullPointerException ex) {

    }
}
```

Question Three

```
@Test
public void test() {
    Assert.assertTrue(V.f(2)>0); // remember f is
        static
    Assert.assertEquals(0.5, V.f(0), 0.001);
    try {
        f(-1);
        Assert.fail();
    } catch (NullPointerException ex) {

    }
}
```

Possible Assumptions:

static import of f from V i.e. `import static V.f;`

static import of Assert.* i.e. `import static Assert.*;`

Question Three

```
@Test
public void test() {
    Assert.assertTrue(V.f(2)>0); // remember f is
        static
    Assert.assertEquals(0.5, V.f(0), 0.001);
    try {
        f(-1);
        Assert.fail();
    } catch (NullPointerException ex) {

    }
}
```

Notes:

assertEquals(0.5, V.f(0)) is not correct.

Catching a broader exception than you need also not correct.

Question Four

Question Four

A Bag is an unordered collection of ints where each value can appear multiple times. For example a bag could contain 1, 5 and seven 3s.

A) What member variables would you need to add to the class?
B) Implement the constructor.
C) Implement add
D) Implement remove
E) Implement getCount

Question Four

A) List: can add and remove items from the end of the list and loop through counting occurences for getCount

B) Set: seems like a good idea but the items in the bag can appear multiple times

C) Map: map values to number of times they appear. Needs filtering to not report on things that were added and then removed

Question Four

A) List: can add and remove items from the end of the list and loop through counting occurences for getCount

B) Set: seems like a good idea but the items in the bag can appear multiple times

C) Map: map values to number of times they appear. Needs filtering to not report on things that were added and then removed

Question Four

A) List: can add and remove items from the end of the list and loop through counting occurences for getCount

B) Set: seems like a good idea but the items in the bag can appear multiple times

C) Map: map values to number of times they appear. Needs filtering to not report on things that were added and then removed

Question Four

A) List: can add and remove items from the end of the list and loop through counting occurences for getCount

B) Set: seems like a good idea but the items in the bag can appear multiple times

C) Map: map values to number of times they appear. Needs filtering to not report on things that were added and then removed