**Chapter 8**

**Recursion Practice**

**(and then there’s more)**

Question 21-22 refer to the method smile below.

21. What is the output when smile(4) is called?

 public static void smile (int n)

 {

 if(n == 0)

 return;

 for (int k=1; k<= n; k++)

 System.out.print(“smile!”);

 smile(n-1);

 }

1. smile!
2. smile!smile!
3. smile!smile!smile!
4. smile!smile!smile!smile!
5. smile!smile!smile!smile!smile!smile!smile!smile!smile!smile!

22. When smile(4) is called, how many times will smile be called including the initial call?

1. 2
2. 3
3. 4
4. 5
5. 10

23. What is displayed when the following method is called with splat(\*\*)?

 public static void splat(String s)

 {

 if(s.length() < 8)

 splat(s+s);

 System.out.println(s);

 }

1. \*\*
2. \*\*\*\*
3. \*\*\*\*\*\*\*\*
4. \*\*\*\*\*\*\*\*

\*\*

1. \*\*\*\*\*\*\*\*

\*\*\*\*

\*\*

24. Lexi is a cheerleader and a programmer. She has written the following recursive mthod that is supposed to generate the cheer “2 4 6 8 who do we appreciate!”:

public void cheer (int i)

 {

 if (i != 8) //line 1

 { //line 2

 i += 2; //line 3

 cheer(i); //line 4

 System.out.print(i + “ “); //line 5

 } //line 6

 else //line 7

 { //line 8

 System.out.print(“who do we appreciate!”); //line 9

 } //line 10

 }

However, Lexi’s method doesn’t work as expected when she calls cheer(0). To get the right cheer, Lexi should

1. replace if (i != 8) with if (i<=8) on line 1
2. replace if (i != 8) with if (i==8) on line 1
3. replace if (i != 8) with while (i!=8) one line 1
4. swap line 4 and line 5
5. move line 3 after line 5

25. Consider the following method:

 public String filter (String str, String pattern)

 {

 int pos = str.indexOf(pattern);

 if (pos ==1)

 return str;

 else

 return filter (str.substring(0,pos) + str.substring(pos+pattern.length()), pattern);

 }

What is the output of

 System.out.print(filter(“papaya”, “pa”));

1. p
2. pa
3. ya
4. aya
5. paya

26. Consider the following method:

 public void mystery (int a, int b)

 {

 System.out.print (a + “ “);

 if (a <= b)

 mystery(a+5, b-1);

 }

 What is the output when mystery(0,16) is called?

1. 0
2. 0 5
3. 0 5 10
4. 0 5 10 15
5. 0 5 10 15 20

27. Consider the following method:

 public int getSomething (int value)

 {

 if (value < 2)

 return 0;

 else

 return 1 + getSomething(value -2);

 }

 Assume val > 0. What is returned by the call getSomething(val)?

1. val – 2
2. val%2
3. (val-1) % 2
4. val /2
5. (val-1) /2

28. Consider the following method:

 public int change (int value)

 {

 if (value < 3)

 return value % 3;

 else

 return value %3 + 10\*change(value/3);

 }

 What will be returned by the call change(45)?

1. 0
2. 21
3. 150
4. 500
5. 1200

29. Consider the following method:

 public void change (int value)

 {

 if (value < 5)

 Sytem.out.print(“” + value%5);

 else

 {

 System.out.print(“” + value%5);

 change(value/5);

 }

 }

 What will be printed as a result of the call change(29)?

1. 1
2. 4
3. 14
4. 104
5. 401

30. Consider the following two methods that are declared within the same class:

 public int supplement (int value)

 {

 if(value < 50)

 return reduce (value + 10);

 else

 return value;

 }

 public int reduce (int value)

 {

 if (value > 0)

 return supplement(value – 5);

 else

 return supplement(value);

 }

What will be returned as a result of the call supplement(40)?

1. 0
2. -5
3. 50
4. 55
5. nothing will be returned due to an infinite recursion