

# Assessment #3 Practice

Multiple Choice - Five questions covering all concepts covered so far!

*skip!*  
*This only works if MAX is multiple of 4*

1. What is printed as a result of executing the following code segment?

```

MAX
int k = 0;
while (k < MAX)
{
    k += 4;
}
System.out.println(k);
    
```

- (A) MAX
- (B) MAX - 4
- (C) MAX + 4
- (D) 4
- (E) Nothing - the program goes into an infinite loop

*MAX → 12*  

<i>k</i>	<i>&lt;</i>	<i>12?</i>
0		✓
4		✓
8		✓
12		no!

*prints 12*

*only works for multiples of 4*

*MAX → 20*  

<i>k</i>	<i>&lt;</i>	<i>20?</i>
0		✓
4		✓
8		✓
12		✓
16		✓
20		stop!

*prints 20*

2. Consider the following method.

```

public int randomPoints(int n)
{
    return (int) (n * Math.random()) + 1;
}
    
```

Which of the following outputs is NOT possible when the statement below is executed?

```

System.out.println(randomPoints(3) + randomPoints(3));
    
```

- (A) 2
- (B) 3
- (C) 4
- (D) 6
- (E) All of the above are possible

*(int) (3 \* math.random()) + 1*  
*produces 1, 2, 3* + *produces 1, 2, 3*  
*1+1 → 2*  
*1+2 → 3*  
*2+2 → 4*  
*3+3 → 6*

3. What is the output of the following code segment?

```

String name = "DonaldDuck";
int pos = name.indexOf("D", 1);
int pos2 = name.indexOf("al");
int pos3 = name.indexOf("x");
System.out.println(pos + pos2 + pos3);
    
```

- (A) 2
- (B) 3
- (C) 4
- (D) 8
- (E) 9

*0 1 2 3 4 5 6 7 8 9*  
*Donald Duck*  
*pos = 6*  
*pos2 = 3*  
*pos3 = -1*  
*6 + 3 + (-1)*

4. Given the declaration

```
int p = 5, q = 3;
```

Which of the following expressions evaluates to 7.5?

- I.  $(\text{double}) (p * q / 2);$   $(\text{double})(5 * 3 / 2) \rightarrow (\text{double})(15 / 2) \rightarrow (\text{double})(7.5)$
- II.  $(\text{double}) p * (\text{double}) q / 2;$   $5.0 * 3.0 / 2 \rightarrow 15.0 / 2 \rightarrow 7.5$
- III.  $(\text{double}) p * (\text{double}) (q / 2);$   $5.0 * (\text{double})(3 / 2)$   
 $5.0 * (\text{double})(1) \rightarrow 6.0$

- (A) I only
- (B) II only
- (C) I and II only
- (D) I, II, and III
- (E) None of them

5. Consider the following code segment.

```
int p = 1;
while (p < 5)
{
    int q = 2;
    while (q < 7)
    {
        q += p;
        p++;
        System.out.println(p + " " + q);
    }
}
```

$p < 5$     $q < 7$

1	✓	2	✓
2	←	3	✓
3	←	5	✓
4		8	✓
-----			
4	✓	2	✓
5	←	6	✓
6		11	no!
6	< 5?	no!	

prints

2 3  
 2 3 3 5 4 8  
 2 3 3 5 4 8 5 6  
 2 3 3 5 4 8 5 6 6 11

What is the last output when the code segment executes?

- (A) 4 5
- (B) 4 8
- (C) 5 6
- (D) 6 10
- (E) 6 11

## Identifying Outputs of "While" Loops

Track the following code segments to determine what they will print.

For problems 1-3, use the following variable declarations:

```
final int MAX = 32;
int num = 15;
```

1. `while (num < MAX)`  
`{`  
`num = num++;`  
`System.out.print(num);`  
`}`

**Output:**

15 16 17 18 19 20 21 22 23 24 25 26 27  
 28 29 30 31 32

2. `while (num < MAX)`  
`{`  
`System.out.println (num);`  
`num *= 2;`  
`}`

num < 32?  
 15    ✓  
 30    ✓  
 60    no!

**Output:**

15  
 30

3. `while (num < MAX)`  
`{`  
`if (num%3 == 0)`  
`System.out.print(num + " ");`  
`num++;`  
`}`

num < 32?    %3?  
 15            ✓  
 16            no  
 17            no  
 18            ✓  
 ↓  
 31            no  
 32 stop

**Output:**

15 18 21 24 27 30

4. `int num = 8905;`  
`int rev = 0;`  
`while (num > 0)`  
`{`  
`rev = rev * 10;`  
`rev = num % 10 + rev;`  
`num = num / 10;`  
`}`

num            rev  
 8905            0    0\*10  
                  0 ← 8905%10 + 0  
                  5    ↓  
 890            5    5\*10  
                  50 ← 890%10 + 50  
                  50 ←  
                  89  
 89            50    50\*10  
                  500 ← 89%10 + 500  
 89/10 ↓    509 ←  
 8            509    509\*10  
                  5090 ← 8%10 + 5090  
 8/10 → 0    5098    ←

**Output:**

System.out.println(rev);

```

5. int num = 5;
   int dum = 4;

   while (num > 4 && dum < 10)
   {
       num++;
       dum = num;
   }

   System.out.print(num + "-" + dum);

```

num > 4	dum < 10	Output:
5 ✓	4 ✓	10-10
6 ✓	6 ✓	
7	7	
8	8	
9	9	
10 ✓	10 no!	

```

6. boolean finished = false;
   int num = 0;
   while (!finished) {
       num += 1;
       if (num % 5 == 0)
           finished = true;
       num += 201;
   }

   System.out.println(num);

```

num	%5 == 0	Output:
0		606
1	φ	
202	φ	
203	φ	
404		
405	✓	→ finished = true
606		stops loop