MCQ: Arrays

This quiz has 14 questions.

1. Consider the following code segment.

```
int[] arr = {1, 2, 3, 4, 5};
```

Which of the following code segments would correctly set the first two elements of array `arr` to `10` so that the new value of array `arr` will be `{10, 10, 3, 4, 5}`?

Ⓐ
```
arr[0] = 10;
arr[1] = 10;
```

Ⓑ
```
arr[1] = 10;
arr[2] = 10;
```

Ⓒ
```
arr[0, 1] = 10;
```

Ⓓ
```
arr[1, 2] = 10;
```

Ⓔ
```
arr = 10, 10, 3, 4, 5;
```

2. Consider the following method.

```
public int[] transform(int[] a) {
    a[0]++;
    a[2]++;
    return a;
}
```

The following code segment appears in a method in the same class as transform.

```
/* missing code */
arr = transform(arr);
```

After executing the code segment, the array `arr` should contain `{1, 0, 1, 0}`. Which of the following can be used to replace `/* missing code */` so that the code segment works as intended?

I. `int[] arr = {0, 0, 0, 0};`

II. `int[] arr = new int[0];`

III. `int[] arr = new int[4];`

Ⓐ I only

Ⓑ II only

Ⓒ III only

Ⓓ I and II

Ⓔ I and III

3. Consider the following method, which is intended to return the number of strings of length greater than or equal to 3 in an array of String objects.

```
public static int checkString(String[] arr)
{
    int count = 0;
    for (int k = 0; k < arr.length; k++) {
        if (arr[k].length() >= 3) {
            count++;
        }
    }
    return count;
}
```

Which of the following code segments compile without error?

I.
```
checkString(new String[]);
```

II.
```
checkString(new String[0]);
```

III.
```
String[] str = {"cat", "dog"};
checkString(str);
```

Ⓐ II only

Ⓑ III only

Ⓒ I and III

Ⓓ II and III

Ⓔ I, II, and III

4. Consider the following code segment.

```
int[] arr = {10, 20, 30, 40, 50};
for(int x=1; x<arr.length-1; x++) {
    arr[x+1] = arr[x] + arr[x+1];
}
```

Which of the following represents the contents of `arr` after the code segment has been executed?

Ⓐ `{10, 20, 30, 70, 120}`

Ⓑ `{10, 20, 50, 90, 50}`

Ⓒ `{10, 20, 50, 90, 140}`

Ⓓ `{10, 30, 60, 100, 50}`

Ⓔ `{10, 30, 60, 100, 150}`

MCQ: Arrays

5. Consider the following code segment.

```
int[] arr = {4, 3, 2, 1, 0};
int total = 0;
for (int k = 0; k <= total; k++) {
    if (arr[k] % 2 == 0) {
        total += arr[k];
    } else {
        total -= arr[k];
    }
}
System.out.print(total);
```

What, if anything, is printed as a result of executing the code segment?

Ⓐ `2`

Ⓑ `1`

Ⓒ `0`

Ⓓ `-4`

Ⓔ Nothing is printed because the code segment causes a runtime error.

6. The array fruits is declared below.

```
String [] fruits = {"apples",
    "bananas", "cherries", "dates"};
```

Which of the following code segments will cause an `ArrayIndexOutOfBoundsException`?

I.
```
for(int i=0; i<=fruits.length; i++)
{
    System.out.println(fruits[i]);
}
```

II.
```
for(int i=0; i<=fruits.length-1; i++)
{
    System.out.println(fruits[i]);
}
```

III.
```
for(int i=1; i<=fruits.length; i++)
{
    System.out.println(fruits[i-1]);
}
```

Ⓐ I only

Ⓑ II only

Ⓒ I and III

Ⓓ II and III

Ⓔ I, II, and III

7. The Fibonacci numbers are a sequence of integers. The first two numbers are 1 and 1. Each subsequent number is equal to the sum of the previous two integers. For example, the first seven Fibonacci numbers are 1, 1, 2, 3, 5, 8, and 13.
The following code segment is intended to fill the fibs array with the first ten Fibonacci numbers. The code segment does not work as intended.

```
int[] fibs = new int[10];
fibs[0] = 1;
fibs[1] = 1;
for (int j=1; j<fibs.length; j++) {
    fibs[j] = fibs[j-2] + fibs[j-1];
}
```

Which of the following best identifies why the code segment does not work as intended?

Ⓐ In the for loop header, the initial value of `j` should be `0`.

Ⓑ In the for loop header, the initial value of `j` should be `2`.

Ⓒ The for loop condition should be `j < fibs.length-1`.

Ⓓ The for loop condition should be `j < fibs.length+1`.

Ⓔ The for loop should increment `j` by `2` instead of by `1`.

MCQ: Arrays

8. Consider the following code segment.

```
int[] numbers = {1, 2, 3, 4, 5, 6};
for (int i=0; i<numbers.length; i++)
{
    System.out.println(numbers[i]);
}
```

Which of the following for loops produces the same output as the code segment?

Ⓐ
```
for (int x : numbers)
{
    System.out.println(numbers[x]);
}
```

Ⓑ
```
for (int x : numbers)
{
    System.out.println(numbers);
}
```

Ⓒ
```
for (int x : numbers)
{
    System.out.println(x);
}
```

Ⓓ
```
for (numbers : int x)
{
    System.out.println(numbers[x]);
}
```

Ⓔ
```
for (numbers : int x)
{
    System.out.println(x);
}
```

9. Consider the following class definition.

Code Segment I

```
public class Toy
{
    private int yearFirstSold;
    public int getYearFirstSold()
    {
        return yearFirstSold;
    }
    /* There may be instance
     * variables, constructors,
     * and other methods not
     * shown.
     */
}
```

The following code segment, which appears in a class other than Toy, prints the year each Toy object in toyArray was first sold by its manufacturer. Assume that toyArray is a properly declared and initialized array of Toy objects.

```
for (Toy k : toyArray)
{
    System.out.println(
        k.getYearFirstSold());
}
```

Which of the following could be used in place of the given code segment to produce the same output?

I.
```
for (int k=0; k<toyArray.length; k++)
{
   System.out.println(
        getYearFirstSold(k));
}
```

II.
```
for (int k=0; k<toyArray.length; k++)
{
    System.out.println(
        k.getYearFirstSold());
}
```

III.
```
for (int k=0; k<toyArray.length; k++)
{
    System.out.println(
        toyArray[k].getYearFirstSold());
}
```

Ⓐ I only

Ⓑ II only

Ⓒ III only

Ⓓ I and II

Ⓔ II and III

MCQ: Arrays

10. Consider the following code segments.

I.
```
int[] arr = {1, 2, 3, 4, 5};
for (int x=0; x<arr.length; x++) {
    System.out.print(arr[x + 3]);
}
```

II.
```
int[] arr = {1, 2, 3, 4, 5};
for (int x : arr) {
    System.out.print(x + 3);
}
```

Which of the following best describes the behavior of code segment I and code segment II ?

Ⓐ Both code segment I and code segment II will print `45`.

Ⓑ Both code segment I and code segment II will print `45678`.

Ⓒ Code segment I will cause an `ArrayIndexOutOfBoundsException` and code segment II will print `45`.

Ⓓ Code segment I will cause an `ArrayIndexOutOfBoundsException` and code segment II will print `45678`.

Ⓔ Both code segment I and code segment II will cause an `ArrayIndexOutOfBoundsException`.

11. The code segment below is intended to set the boolean variable duplicates to true if the int array arr contains any pair of duplicate elements. Assume that arr has been properly declared and initialized.

```
boolean duplicates = false;
for (int x=0; x<arr.length-1; x++) {
    /* missing loop header */
    {
        if (arr[x] == arr[y]) {
            duplicates = true;
        }
    }
}
```

Which of the following can replace `/* missing loop header */` so that the code segment works as intended?

Ⓐ `for(int y=0; y<=arr.length; y++)`

Ⓑ `for(int y=0; y<arr.length; y++)`

Ⓒ `for(int y=x; y<arr.length; y++)`

Ⓓ `for(int y=x+1; y<arr.length; y++)`

Ⓔ `for(int y=x+1;y<=arr.length; y++)`

12. In the code segment below, assume that the int array `numArr` has been properly declared and initialized. The code segment is intended to reverse the order of the elements in numArr. For example, if `numArr` initially contains `{1, 3, 5, 7, 9}`, it should contain `{9, 7, 5, 3, 1}` after the code segment executes.

```
/* missing loop header */
{
    int temp=numArr[k];
    numArr[k]=numArr[numArr.length-k-1];
    numArr[numArr.length-k-1]=temp;
}
```

Which of the following can be used to replace /* missing loop header */ so that the code segment works as intended?

Ⓐ `for (int k=0; k<numArr.length/2; k++)`

Ⓑ `for (int k=0; k<numArr.length; k++)`

Ⓒ `for (int k=0; k<numArr.length/2; k--)`

Ⓓ `for (int k=numArr.length-1; k>=0; k--)`

Ⓔ `for (int k=numArr.length-1; k>=0; k++)`

13. Consider the following code segment, which is intended to print the maximum value in an integer array values. Assume that the array has been initialized properly and that it contains at least one element.

```
int maximum=/*missing initial value*/;
for (int k=1; k<values.length; k++) {
    if (values[k] > maximum) {
        maximum = values[k];
    }
}
System.out.println(maximum);
```

Which of the following should replace `/*missing initial value*/` so that the code segment will work as intended?

Ⓐ `0`

Ⓑ `values[0]`

Ⓒ `values[1]`

Ⓓ `Integer.MIN_VALUE`

Ⓔ `Integer.MAX_VALUE`

MCQ: Arrays

14. Consider the following method, which is intended to return the index of the first negative integer in a given array of integers.

```
public int posOfFirstNeg(int[] values)
{
    int index = 0;
    while (values[index] >= 0) {
        index++;
    }
    return index;
}
```

What precondition is needed on the values array so that the method will work as intended?

Ⓐ The array values must contain at least one negative integer.

Ⓑ The array values must contain at least one nonnegative integer.

Ⓒ The array values must contain at least one positive integer.

Ⓓ No precondition is needed. The method will never work as intended.

Ⓔ No precondition is needed. The method will always work as intended.