Module 5 Quiz

LATEST SUBMISSION GRADE

100%

1. Of the following, which is the correct syntax to declare an array of 24 **boolean** elements named **arr**?

1 / 1 point

● boolean[] arr = new boolean[24];

○ boolean[24] arr = new boolean[24];

○ boolean
arr = new boolean[24];

○ boolean arr[] = new boolean;

○ boolean arr[24];

✓ **Correct**

2. Given the declaration of the 24 element **boolean** array **arr** above, what is the index of the last element of the array?

1 / 1 point

○ 24

○ 25

○ 22

● 23

✓ **Correct**

Java arrays use zero-based
indexing, where the index of the first array element is zero.

3. If you wanted to initialize the **boolean** array **arr** so that all elements contain the value **true**, you could use the following: **arr.fill(true);**

1 / 1 point

● FALSE

○ TRUE

✓ **Correct**

Array objects do not have
many methods, and in particular they do not have a fill() method. Please review
the methods available in the Arrays class.

4. Arrays of class type require a two-phase initialization. One phase to create the array and another phase to create the objects in the array.

1 / 1 point

○ FALSE

● TRUE

✓ **Correct**

Arrays of class type only
contain references to the actual objects.

5. Given an integer array named **arr**, the number of elements in the array can be determined by the following method call: **arr.length()**

1 / 1 point

● FALSE

○ TRUE

✓ **Correct**

Please review what methods
are available in array objects and what data fields are available in them.

6. Which of the following statements is true (pick only one):

1 / 1 point

○ An array can be sized dynamically, and it can be resized without allocating a new array.

● An array can be sized dynamically, but it cannot be resized without allocating a new array.

○ An array cannot be sized dynamically when the program is running.

✓ **Correct**

The size of an array is set
at runtime but then it is fixed.

7. What is the value of the variable s after the following section of code executes?

1 / 1 point

```
int s = 0;
int [] a = {12, 23, 34, 45, 56};
for (int i=1; i<a.length; i++) {
   s += a[i];
}

```

○ 114

● 158

○ 170

○ 0

✓ **Correct**

Carefully examine the for-loop and where it starts and stops.

8. What type of collection would we use if we wanted no duplicates?

1 / 1 point

○ Map

○ Queue

● Set

○ List

✓ **Correct**

Please review the various containers available in the Java Collections Framework in lecture 4 part 1.

9. Examine the following code:

1 / 1 point

```
ArrayList<String> list = new ArrayList<String>() ;

list.add("alpha");
list.add("bravo");
list.add("charlie");
list.add("delta");
list.add("echo");

```

Which of the following will replace the element "**charlie**" with "**foxtrot**"?

○ list.set("foxtrot","charlie");

○ list.add("foxtrot", list.indexOf("charlie"));

● list.set(list.indexOf("charlie"),"foxtrot");

○ list.set("charlie","foxtrot");

○ list[2] = "foxtrot";

✓ **Correct**

In this situation, we want to replace an element rather than add a new element. Review the methods of the ArrayList class in lecture 5.

10. Examine the following code:

1 / 1 point

```
ArrayList<String> list = new ArrayList<String>() ;

list.add("alpha");
list.add("bravo");
list.add("charlie");
list.add("delta");
list.add("echo");

```

Which of the following will change the list so that it looks like:

```
alpha
bravo
charlie
delta
```

○ list.remove(list.size());

○ list.remove(5);

○ list.clear("echo");

● list.remove(list.size()-1);

○ list.empty("echo");

✓ **Correct**

In this case we want to delete an element from the
ArrayList. There are several ways to do it. Review the methods available in the
class.

11. Examine
the following code:

1 / 1 point

```
ArrayList<String> list = new ArrayList<String>() ;

list.add("alpha");
list.add("bravo");
list.add("charlie");
list.add("delta");
list.add("echo");

for ( ___________ name : ___________ ) {
    out.println( ______________ );
}

```

Fill in the blanks so that all the elements in the ArrayList are printed.

○
```
int   String   name
```

●
```
String   list   name
```

○
```
String   iterator()   next()
```

○
```
String   list   next()
```

○
```
iterator()   hasNext()   next()
```

✓ **Correct**

Please review the syntax of the foreach loop in lecture
3

12. A HashMap can map keys of any type to values of any type.

1 / 1 point

○ TRUE

● FALSE

✓ **Correct**

The containers in the Java
Collections Framework only store data that are object types. Primitive type
data requires the usage of a wrapper class instead.

13. Which of the following is ***not*** a feature/advantage of a HashMap?

1 / 1 point

○ HashMaps can use any object type as a key.

○ HashMaps have fast lookup.

○ HashMaps have fast insert.

● HashMaps keep their keys in sorted order.

✓ **Correct**