Section 8 Quiz

(Answer all questions in this section)

1. What is the output?

```
int[] arr = new int[1];
arr[0] = 10;
System.out.println(arr[0]);
```

Mark for Review (1) Points

- 10 (*)
- ArrayIndexOutOfBoundsException
- 0
- 1

Incorrect. Refer to Section 8 Lesson 1.

2. You can access the size of any array by using the array's "length" property.

Mark for Review (1) Points

- True (*)
- False

Incorrect. Refer to Section 8 Lesson 1.

3. Which loop type is specially designed to traverse an array?

Mark for Review (1) Points

- do while loop
- repeat loop
- while loop
- for loop (*)

Incorrect. Refer to Section 8 Lesson 1.

4. What is the starting index of an array?

Mark for Review (1) Points

- 0 (*)
- It depends on the type of the array.
- You can start with anything
- 1

Incorrect. Refer to Section 8 Lesson 1.

5. Arrays are like variables which must be declared prior to use.

Mark for Review (1) Points

- True (*)
- False

Incorrect. Refer to Section 8 Lesson 1.

6. The Java compiler does not check for an ArrayIndexOutOfBoundsException during the compilation of a program containing arrays.

Mark for Review (1) Points

- True (*)
- False

Incorrect. Refer to Section 8 Lesson 1.

7. How could you declare an ArrayList so that it can store true or false values?

Mark for Review (1) Points

- ArrayList<True, False> arrList = new ArrayList<>();
- ArrayList<boolean> arrList = new ArrayList<>();
- ArrayList<true, false> arrList = new ArrayList<>();
- ArrayList<Boolean> arrList = new ArrayList<>(); (*)

Incorrect. Refer to Section 8 Lesson 2.

8. You can access elements in an ArrayList by their index.

Mark for Review (1) Points

- True (*)
- False

Incorrect. Refer to Section 8 Lesson 2.

9. Which is not used to traverse an ArrayList?

Mark for Review (1) Points

- ListIterator
- iterator
- for-each loop
- do- while loop (*)

Incorrect. Refer to Section 8 Lesson 2.

10. Which exception is thrown when an application attempts to use null when an object is required?

Mark for Review (1) Points

- NullPointerException (*)
- FileNotFoundException
- ArithmeticException
- ArrayIndexOutOfBoundsException

Correct

11. What happens when you don't handle an exception?

Mark for Review (1) Points

- (•) The execution of the program is terminated abruptly. (*)
- ( ) The program encounters error and simply ignores it.
- ( ) A message is printed to the console to ask you how to handle the error.
- ( ) All of the code after the error is skipped, but the program still runs.

Correct

12. Which code goes in the try block?

Mark for Review
(1) Points

- ( ) Any code that is likely to cause an exception. (*)
- (•) Any code that is safe from an exception.
- ( ) Any code that can handle an exception.
- ( ) Any code that is likely to print the exception details.

Incorrect. Refer to Section 8 Lesson 3.

13. Runtime errors can be caught by Java's exception handling mechanism.

Mark for Review
(1) Points

- ( ) True (*)
- (•) False

Incorrect. Refer to Section 8 Lesson 4.

14. Using the NetBeans debugger, you can set breakpoints and trace through a program one line at a time.

Mark for Review
(1) Points

- ( ) True (*)
- (•) False

Incorrect. Refer to Section 8 Lesson 4.

15. Identify where there is a potential bug in this code:

```
int radiusOfCircle = 10;
int areaOfCircle = Math.PI * radiusOfCircle * radiusOfCircle;
```

Mark for Review
(1) Points

- (•) A datatype is incorrect. (*)
- ( ) A variable hasn't been assigned a value.
- ( ) A semi-colon is missing.
- ( ) A variable name is misspelled.

Correct

1. Arrays are like variables which must be declared prior to use.

Mark for Review
(1) Points

- ( ) True (*)

- False

Incorrect. Refer to Section 8 Lesson 1.

2. Given:
int x[];
What is the value of x?

Mark for Review
(1) Points

- 1
- 0
- Some random number.
- null (*)

Correct

3. You can access the size of any array by using the array's "length" property.

Mark for Review
(1) Points

- True (*)
- False

Incorrect. Refer to Section 8 Lesson 1.

4. The Java compiler does not check for an ArrayIndexOutOfBoundsException during the compilation of a program containing arrays.

Mark for Review
(1) Points

- True (*)
- False

Incorrect. Refer to Section 8 Lesson 1.

5. What is the output?
int[] arr = new int[1];
arr[0] = 10;
System.out.println(arr[0]);

Mark for Review
(1) Points

- 0
- 1
- 10 (*)
- ArrayIndexOutOfBoundsException

Incorrect. Refer to Section 8 Lesson 1.

6. Which two are valid array declarations?

Mark for Review
(1) Points

(Choose all correct answers)

- [ ] int array size;
- [x] int[] size; (*)
- [ ] []int size;
- [ ] int size[]; (*)

Incorrect. Refer to Section 8 Lesson 1.

7. Which of the following is not a wrapper class?

Mark for Review
(1) Points

- Boolean
- String (*)
- Integer
- Byte

Incorrect. Refer to Section 8 Lesson 2.

8. You can access elements in an ArrayList by their index.

Mark for Review
(1) Points

- True (*)
- False

Incorrect. Refer to Section 8 Lesson 2.

9. Which is NOT a benefit of ArrayList class?

Mark for Review
(1) Points

- You can remove all of the elements of an ArrayList with a method.
- You can use an ArrayList list to store Java primitive values (like int). (*)
- An ArrayList grows as you add elements.
- An ArrayList shrinks as you remove elements.

Incorrect. Refer to Section 8 Lesson 2.

10. Identify where there is a potential bug in this code:

Mark for Review
(1) Points

```
int radiusOfCircle = 10;
int areaOfCircle = Math.PI * radiusOfCircle * radiusOfCircle;
```

- A semi-colon is missing.
- A variable hasn't been assigned a value.
- A datatype is incorrect. (*)
- A variable name is misspelled.

Correct

11. Testing and debugging are important activities in software development.

Mark for Review
(1) Points

- True (*)
- False

Incorrect. Refer to Section 8 Lesson 4.

12. Which is not a compilation error?

Mark for Review
(1) Points

- int y;
  y++; (*)
- y = 3 + * 5;
- x = (3 + 5;
- int x=2

Incorrect. Refer to Section 8 Lesson 4.

13. What is the danger of catching a generic Exception type as shown below?

Mark for Review
(1) Points

```
int[] array = {10, 20, 30};
int b = 0;
try{
   System.out.println("1");
   int c = (array[3] / b);
   System.out.println("2");
}
catch(Exception ex){
   System.out.println(ex.toString());
}
```

- The details of the Exception object ex are too general to be useful. (*)
- An Exception will never occur.
- An ArithmeticException cannot be caught.
- An ArrayIndexOutOfBoundsException cannot be caught.

Incorrect. Refer to Section 8 Lesson 3.

14. What is the output?

Mark for Review
(1) Points

```
int[] array = {10, 20, 30};
int b = 0;
try{
   System.out.println("1");
   int c = (array[3] / b);
   System.out.println("2");
}
catch(ArithmeticException ex){
   System.out.println("Arithmetic Exception");
}
catch(ArrayIndexOutOfBoundsException ex){
   System.out.println("Array index out of bounds");
}
```

- 1
  Arithmetic Exception
- 1
  Array index out of bounds (*)
- 1
  2
- Array index out of bounds
- 1
  2
  Array index out of bounds

Incorrect. Refer to Section 8 Lesson 3.

15. Which exception is thrown when an application attempts to use null when an object is required?

Mark for Review
(1) Points

- NullPointerException (*)
- FileNotFoundException
- ArithmeticException
- ArrayIndexOutOfBoundsException

Incorrect. Refer to Section 8 Lesson 3.

1. The size of an ArrayList can grow as needed.

Mark for Review
(1) Points

- True (*)
- False

Incorrect. Refer to Section 8 Lesson 2.

2. A wrapper class encapsulates, or wraps, the primitive types within an object.

Mark for Review
(1) Points

- True (*)
- False

Incorrect. Refer to Section 8 Lesson 2.

3. Which is NOT a benefit of ArrayList class?

Mark for Review
(1) Points

- You can use an ArrayList list to store Java primitive values (like int). (*)
- An ArrayList shrinks as you remove elements.
- You can remove all of the elements of an ArrayList with a method.
- An ArrayList grows as you add elements.

Incorrect. Refer to Section 8 Lesson 2.

4. Which loop type is specially designed to traverse an array?

Mark for Review
(1) Points

- repeat loop
- do while loop
- for loop (*)
- while loop

Incorrect. Refer to Section 8 Lesson 1.

5. What is the output?

```
int[] arr = new int[1];
arr[0] = 10;
System.out.println(arr[0]);
```

Mark for Review
(1) Points

- 1
- 0
- 10 (*)

ArrayIndexOutOfBoundsException

Incorrect. Refer to Section 8 Lesson 1.

6. What is the starting index of an array?

Mark for Review
(1) Points

- It depends on the type of the array.
- 1
- You can start with anything
- 0 (*)

Incorrect. Refer to Section 8 Lesson 1.

7. What is the output?

```
int[] arr = new int[5];
for(int i=0; i<arr.length; i++){
   arr[i] = i;
}
for(int i=0; i<arr.length; i++) {
   System.out.print(arr[i]);
}
```

Mark for Review
(1) Points

- 12345
- 012345
- 01234 (*)
- 123

Incorrect. Refer to Section 8 Lesson 1.

8. What is an array?

Mark for Review
(1) Points

- An array is an indexed container that holds a set of values of a multiple types.
- An array is a Java primitive type.
- An array is a way to create multiple copies of a single value.
- An array is an indexed container that holds a set of values of a single type. (*)

Incorrect. Refer to Section 8 Lesson 1.

9. Arrays are like variables which must be declared prior to use.

Mark for Review
(1) Points

- True (*)
- False

Incorrect. Refer to Section 8 Lesson 1.

10. What is the output?

```
int[] array = {10, 20, 30};
int b = 0;
try{
```

Mark for Review
(1) Points

```
    System.out.println("1");
    int c = (array[3] / b);
    System.out.println("2");
}
catch(ArithmeticException ex){
    System.out.println("Arithmetic Exception");
}
catch(ArrayIndexOutOfBoundsException ex){
    System.out.println("Array index out of bounds");
}
```

- 1
  Arithmetic Exception
- 1
  Array index out of bounds (*)
- 1
  2
- Array index out of bounds
- 1
  2
  Array index out of bounds

Incorrect. Refer to Section 8 Lesson 3.

11. What happens when you don't handle an exception?

Mark for Review
(1) Points

- The execution of the program is terminated abruptly. (*)
- The program encounters error and simply ignores it.
- A message is printed to the console to ask you how to handle the error.
- All of the code after the error is skipped, but the program still runs.

Incorrect. Refer to Section 8 Lesson 3.

12. If the try block succeeds then no exception has occurred.

Mark for Review
(1) Points

- True (*)
- False

Incorrect. Refer to Section 8 Lesson 3.

13. Which is not a compilation error?

Mark for Review
(1) Points

- int y;
  y++; (*)
- int x=2
- x = (3 + 5;
- y = 3 + * 5;

Incorrect. Refer to Section 8 Lesson 4.

14. Runtime errors can be caught by Java's exception handling mechanism.

Mark for Review
(1) Points

- True (*)
- False

Incorrect. Refer to Section 8 Lesson 4.

15. Testing and debugging are important activities in software development.

Mark for Review
(1) Points

- True (*)
- False

Incorrect. Refer to Section 8 Lesson 4.