

## Quiz Submissions - Java Review Homework Quiz



Chork Hieng (username: gt9182iu)

### Retaken Attempt 2

Written: Jan 21, 2022 1:43 AM - Jan 21, 2022 1:46 AM

### Submission View

Your quiz has been submitted successfully.

Show All Questions

### Java Review

**Question 1** Correct on previous attempt(s) 1 / 1 point

Write a line of code that declares an integer variable called **answer**, and initializes it with the value 42.

Answer: `int answer = 42;` ✓

→ **Question 2** Retaken 0 / 1 point

Write a line of code that declares a floating point variable called **closeToPi**, and initializes it with the value 3.14159.

Answer: `doube closeToPi = 3.14159;` ✗ `(/double +closeToPi *= *3.14159 */)`

**Question 3** Correct on previous attempt(s) 1 / 1 point

Write a line of code that declares a boolean variable called **done**, and initializes it with the value **false**.

Answer: `boolean done = false;` ✓

→ **Question 4** Retaken 1 / 1 point

Write a line of code that declares a floating point variable called **closeToPi**, and initializes it with the result of dividing 22 by 7. Express 22 and/or 7 as floating point numbers to ensure the division has a floating point result. (Do not use an explicit cast of an **int** to a **double**.)

Answer: `double closeToPi = 22.0/7.0;` ✓

**Question 5** Correct on previous attempt(s) 1 / 1 point

Write a line of code that declares a boolean variable called **xIsSmaller** to the result of checking to see if **x** is less than **y**. (Assume **x** and **y** are already declared and initialized.)

Do not use an **if** statement.

Answer: `boolean xIsSmaller = x < y;` ✓

▶ View Feedback

---

**Question 6** Correct on previous attempt(s)

1 / 1 point

Write the first line of an **if** statement that checks to see if an integer variable called **num** is greater than 42.

Answer: `if(num > 42)` ✓

---

→ **Question 7** Retaken

1 / 1 point

Write the first line of a **while** loop that runs as long as a boolean variable called **done** is false. (NOTE: Don't use **done == false**; use the **!** operator.)

Answer: `while(!done)` ✓

---

**Question 8** Correct on previous attempt(s)

1 / 1 point

Write the first line of a **while** loop that executes as long as **lo** is less than **hi**.

Answer: `while(lo < hi)` ✓

---

**Question 9** Correct on previous attempt(s)

1 / 1 point

Write the first line of a **for** loop that iterates over array **a**. Use **i** as the loop variable.

Answer: `for (int i = 0; i < a.length; i++)` ✓

---

**Question 10** Correct on previous attempt(s)

1 / 1 point

What will be the output of this code?

```
int answer = 42;
System.out.print(answer++);
System.out.print(" ");
System.out.print(++answer);
System.out.print(" ");
System.out.print(answer);
```

Answer: 42 44 44 ✓

What will be the output of this code?

```
int[] primes = { 2, 3, 5, 7, 11, 13, 17, 19 };
int lo = 2;
int hi = 6;
System.out.print(primes[lo++]);
System.out.print(" ");
System.out.print(primes[--hi]);
```

Answer: 5 13 ✓

Consider the following code:

```
Scanner sc = new Scanner(System.in);
while (sc.hasNext()) {
    String s = sc.next();
    if (s.equals(" "))
        continue;
    if (s.equals("end"))
        break;
    executeCommand(s);
}
sc.close();
```

If the continue statement is executed, what happens next?

- ✓ ☒ `sc.hasNext()` is called.
- ☐ `executeCommand()` is called.
- ☐ `sc.close()` is called.
- ☐ The **break** statement is executed.

▶ View Feedback

Consider the following code:

```
Scanner sc = new Scanner(System.in);
while (sc.hasNext()) {
```

```
String s = sc.next();
if (s.equals(" "))
    continue;
if (s.equals("end"))
    break;
executeCommand(s);
}
sc.close();
```

If the **break** statement is executed, what happens next?

- ☒ **sc.close()** is called.
- ☐ **sc.hasNext()** is called.
- ☐ **executeCommand()** is called.
- ☐ The **continue** statement is executed.

▶ View Feedback

---

→ Question 14 Retaken

1 / 1 point

Which **while** loop is equivalent to this **for** loop?

```
for (int i = 0; i < a.length; i++)
    sum += a[i];
```

Assume sum and array a are declared.

- ☐

```
while (i < a.length) {
    int i = 0;
    sum += a[i];
    i++;
}
```
- ☐

```
int i = 0;
while (i < a.length) {
    i++;
    sum += a[i] ;
}
```
- ☐

```
int i = 0;
while (i < a.length) {
    sum += a[i];
}
i++;
```



```
int i = 0;
while (i < a.length) {
    sum += a[i];
    i++;
}
```

▶ View Feedback

---

**Question 15** Correct on previous attempt(s)

1 / 1 point

What is wrong with this code snippet? (Note: a syntax error will cause the code to never run.)

```
int[10] a;
a[0] = 42;
```

- ☐ The code is fine as is.
- ☐ 0 is never a valid array index.
- ☐ The array is never allocated, so a[0] references non-existent memory.
- ☒ The array is declared incorrectly.

---

→ **Question 16** Retaken

1 / 1 point

What is wrong with this code snippet? (Note: a syntax error will cause the code to never run.)

```
int[] a;
a[0] = 42;
```

- ☐ The array is declared incorrectly.
- ☒ The array is never allocated, so a[0] references non-existent memory.
- ☐ 0 is never a valid array index.
- ☐ The code is fine as is.

▶ View Feedback

---

**Question 17** Correct on previous attempt(s)**1 / 1 point**

What is wrong with this code snippet? (Note: a syntax error will cause the code to never run.)

```
int[] a = new int[10];  
a[0] = 42;
```

- ☐ The array is never allocated, so a[0] references non-existent memory.
- ☐ The array is declared incorrectly.
- ☐ 0 is never a valid array index.
- ✓ ☒ The code is fine as is.

▶ View Feedback

---

**→ Question 18** Retaken**1 / 1 point**

What is wrong with this code snippet? (Note: a syntax error will cause the code to never run.)

```
int[] a = { 27, 82, 41, 124, 62 };  
int last = a[5];
```

- ✓ ☒ 5 is not a valid array index.
- ☐ The code is fine as is.
- ☐ The array is declared incorrectly.
- ☐ The array is never allocated.

▶ View Feedback

---

**Question 19** Correct on previous attempt(s)**1 / 1 point**

Is this code valid?

```
public static int find(int[] a, int lo, int hi, int value) {  
    for (int i = lo; i < hi; i++)
```

```
        if (a[i] == value)
            return i;
    return -1;
}
public static int find(int[] a, int value) {
    return find(a, 0, a.length, value);
}
```

- ☐ No. The first find() function has two return statements.
- ☐ No. There needs to be a blank line between the functions.
- ☒ Yes, this is valid code.
- ☐ No. There are two functions with the same name.

 [View Feedback](#)

---

**Attempt Score:** 18 / 19 - 94.74 %

**Overall Grade (highest attempt):** 18 / 19 - 94.74 %

Done