

Java Programming 1 - Quiz Notes

Kensukeken

October 9th, 2024

Contents

1	Calculating Powers in Java	2
2	Math.abs Function	2
3	Reading an Integer from User Input	2
4	Declaring Constants	2
5	Output of Floating Point Division	2
6	Modulo Operation	3
7	Output Example	3
8	Math.floor Function	3
9	Integer Division	3
10	Storing Decimal Numbers	3
11	Comparing Numbers	4
12	For Loop Output	4
13	Do-While Loop Output	4
14	If-Else Statement	4
15	Nested Loops Output	5
16	Output of the while loop	5
17	Switch Case Example	5
18	Generating a random integer between 1 and 10:	6

1 Calculating Powers in Java

In Java, you can calculate 2^8 using the `Math.pow` method:

```
double result = Math.pow(2, 8);  
System.out.println(result); // Outputs 256.0
```

Alternatively, using bitwise shift operators:

```
int result = 1 << 8;  
System.out.println(result); // Outputs 256
```

2 Math.abs Function

The result of `Math.abs(7 - 9)` is:

$$7 - 9 = -2$$

Applying `Math.abs` gives 2.

3 Reading an Integer from User Input

To read an integer from user input in Java, the following steps are required:

- Import the `Scanner` class:

```
import java.util.Scanner;
```

- Create a `Scanner` object:

```
Scanner input = new Scanner(System.in);
```

- Use the `nextInt` method to read an integer:

```
int value = input.nextInt();
```

4 Declaring Constants

To declare a constant `HST` in Java, use the `final` keyword:

```
final double HST = 13.0;
```

5 Output of Floating Point Division

The result of `System.out.println(4.0 / 2)` is:

$$4.0/2 = 2.0$$

6 Modulo Operation

The result of `System.out.println(7 % 2)` is:

$$7 \div 2 = 3 \quad (\text{quotient}), \quad 1 \quad (\text{remainder})$$

Thus, $7\%2 = 1$.

7 Output Example

To output the following:

```
Your numbers: 1 2 3 4 5
Total 15
```

The correct Java code is:

```
System.out.println("Your numbers:");
System.out.print(1 + " ");
System.out.print(2 + " ");
System.out.print(3 + " ");
System.out.print(4 + " ");
System.out.print(5 + "\n");
System.out.print("Total ");
System.out.println(15);
```

8 Math.floor Function

The result of `Math.floor(3.9)` is 3.0.

9 Integer Division

The result of `System.out.println(7 / 2)` is 3 because Java performs integer division, truncating the decimal.

10 Storing Decimal Numbers

To store a decimal number like 7.5 in Java, use the `double` type:

```
double value = 7.5;
```

11 Comparing Numbers

The following comparisons are evaluated as:

- $-2 < 1$: True
- $7 < -8$: False
- $-4 < -5$: False
- $3 < 3$: False

12 For Loop Output

For the following code:

```
for (int i = 0; i <= 10; i++) {  
    System.out.println(i);  
}
```

The output is:

0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10

13 Do-While Loop Output

For the following code:

```
int x = 10;  
do {  
    System.out.println(--x);  
} while (x < 0);
```

The output is 9 because the condition is checked after the first iteration.

14 If-Else Statement

Given the code:

```
if (x >= 2) {  
    System.out.println(">=2");  
} else if (x >= 4) {  
    System.out.println(">=4");  
} else {  
    System.out.println("<2");  
}
```

If $x = 4$, the output will be ≥ 2 .

15 Nested Loops Output

For the following code:

```
for (int i = 0; i < 3; i++) {  
    for (int j = 0; j < 3; j++) {  
        System.out.println(j + " - " + i);  
    }  
}
```

The output will be:

0 - 0, 1 - 0, 2 - 0, 0 - 1, 1 - 1, 2 - 1, 0 - 2, 1 - 2, 2 - 2

16 Output of the while loop

```
int x = 0;  
while (x < 10) {  
    System.out.println(++x);  
}  
// Output:  
1  
2  
3  
4  
5  
6  
7  
8  
9  
10
```

17 Switch Case Example

For the following code:

```
int x = 3;  
switch(x) {  
    case 1: System.out.println("one");  
    case 2: System.out.println("two");  
    case 3: System.out.println("three");  
    case 4: System.out.println("four");  
    default: System.out.println("other");  
}  
// Output:  
three  
four  
other
```

18 Generating a random integer between 1 and 10:

```
import java.util.Random;
Random random = new Random();
int randomValue = random.nextInt(10) + 1;
```

Break and Continue Statements

The following code demonstrates the use of **break** and **continue**:

```
for (int i = 0; i < 5; i++) {
    if (i == 2) {
        continue;
    }
    if (i == 4) {
        break;
    }
    System.out.println(i);
}
```

The output is:

0, 1, 3

Ternary Operator

In Java, the ternary operator is used for conditional expressions:

```
int result = (x > 0) ? x : -x;
```

If **x** is greater than 0, **result** is **x**; otherwise, **result** is **-x**.