

# Data Types and Operators

## Question 1

---

Take three inputs from the user.

1. Divide the first number by three and store the remainder in a variable. Then print the variable. (Be careful about the data type)
2. What is the quotient after dividing the first number by the second number? Store the quotient in a variable and print it.
3. The quotient you got from question b, add the third number and store another variable to print it.

### Code

```
import java.util.Scanner;
class assignment {
    public static void main(String args[]) {
        Scanner inputs = new Scanner(System.in);
        int inputX = inputs.nextInt();
        int inputY = inputs.nextInt();
        int inputZ = inputs.nextInt();
        double taskX = inputX / 3;
        int taskY = inputX % inputY;
        int taskZ = taskY + inputZ;
        System.out.println(taskX);
        System.out.println(taskY);
        System.out.println(taskZ);
    }
}
```

### Example

```
6
7
3
2.0
6
9
```

## Question 2

---

What's the difference between increment and decrement operator? Give one example of each type. Can we use increment and decrement operators with character variables? Discuss with a suitable model.

### Code

```
import java.util.Scanner;
class assignment {
    public static void main(String args[]) {
        int intX = 28;
        intX ++;
        System.out.println(intX);
        int intY = 28;
        intY --;
        System.out.println(intY);
        char charX = 'B';
        charX ++;
        System.out.println(charX);
        char charY = 'B';
        charY --;
        System.out.println(charY);
    }
}
```

## Output

```
29
27
C
A
```

## Explanation

`++` increments by one:  $28 + 1 = 29$ .

`--` decrements by one:  $28 - 1 = 27$ .

When incrementing a character, the character changes to the next character of the alphabet. When decrementing a character, the character changes to the last character of the alphabet. This is because characters are stored as `ASCII` in Java. Characters are assigned numbers.

## Question 3

What are compound assignment operators? About how many compound assignments do you know? Can you give one code example of each type? What happens when you use the division assignment operator for zero value ( $a /= 0$ )?

## Description

Compound assignment operators assign the output of their operation. I have learnt five of them. See my code example below. When we use division assignment by zero, we get the output:

```
Exception in thread "main" java.lang.ArithmeticException: / by zero at assignment.main(assignment.java:5) .
```

## Code

```
import java.util.Scanner;
class assignment {
    public static void main(String args[]) {
        int a = 5;
        a += 3;
        System.out.println("Addition assignment: " + a);
        int b = 10;
        b -= 6;
        System.out.println("Subtraction assignment: " + b);
        int c = 20;
        c *= 5;
        System.out.println("Multiplication assignment: " + c);
        int d = 40;
        d /= 10;
        System.out.println("Division assignment: " + d);
        int e = 80;
        e %= 7;
        System.out.println("Modulus assignment: " + e);
    }
}
```

## Output

```
Addition assignment: 8
Subtraction assignment: 4
Multiplication assignment: 100
Division assignment: 4
Modulus assignment: 3
```

## Question 4

---

Name two relational operators we can use in any type in Java, and discuss how we can use those two relational operators with a code example.

### Answer

More than and not equal to.

### Example

```
import java.util.Scanner;
class assignment {
    public static void main(String args[]) {
        int a = 7;
        boolean x = a > 5;
        boolean y = a > 10;
        int b = 20;
        boolean p = b != 20;
        boolean q = b != 40;
        System.out.println(x);
        System.out.println(y);
        System.out.println(p);
        System.out.println(q);
    }
}
```

## Output

```
true
false
false
true
```

## Discussion

`a` is more than five. `a > 5` is `true`. Hence, the boolean `x` is `true`. `a` is not more than five. `a > 10` is `false`. Hence, the boolean `y` is `false`. `b` is equal to twenty. `b != 20` is `false`. Hence, the boolean `p` is `false`. `b` is not equal to forty. `b != 40` is `true`. Hence, the boolean `q` is `true`.

## Question 5

Write down the most used Boolean logical operators in Java. What is the difference between logical AND and short-circuit AND?

## Answer

The boolean logical operators I have learnt about in Java are presented in the table below.

Operator	Function
<code>&amp;</code>	<code>and</code>
<code> </code>	<code>or</code>
<code>^</code>	exclusive <code>or</code>
<code>!</code>	not
<code>==</code>	equal to
<code>!=</code>	not equal to
<code>&amp;&amp;</code>	short-circuit <code>and</code>
<code>  </code>	short-circuit <code>or</code>

Short-circuit `AND` does not check for the state of the second value when the first value is `false`, whereas logical `AND` checks for both values regardless of the state of the first value.