

# Fundamental Programming Structures in Java

## IFT 194: HW 2

Brandon Doyle  
[bdoyle5@asu.edu](mailto:bdoyle5@asu.edu)  
1215232174

Dr. Usha Jagannathan  
[Usha.Jagannathan@asu.edu](mailto:Usha.Jagannathan@asu.edu)

July 11, 2018

## Section 2.1

### 2.1

The difference between the expressions `4`, `4.0`, `'4'`, and `"4"` is the type. For instance, the first is an `int`, the second may be a `double` or `float`, the third is a `char`, and the fourth is a `String` object.

### 2.3

We next examine the output of the following code fragment.

```
System.out.print("Here we go!");  
System.out.println("12345");  
System.out.print("Test this if you are not sure.");  
System.out.print("Another.");  
System.out.println();  
System.out.println("All done.");
```

We expect this to output the following.

```
Here we go!12345  
Test this if you are not sure.Another.  
All done.
```

This is purely a result of using `println` as opposed to just `print`. Moreover, `println` can be called without any arguments.

### 2.4

We're asked to determine what is wrong with the following statement.

```
System.out.println("To be or not to be, that is the question.");
```

I honestly don't think there could be anything wrong with this statement, unless it's pointing at the fact that we may not continue strings on more than one line (as it may be formatted in the text). In that case, it is required that we use concatenation as

```
System.out.println("To be or not to be, that"  
    + " is the question");
```

### 2.8

We are asked to determine which value is contained in the primitive `int` variable `depth` after the following lines execute.

---

View the source of this document on [GitHub](#).

```

size = 18;
size = size + 12;
size = size * 2;
size = size / 4;

```

These statements evaluate to

$$\frac{2(18 + 12)}{4} = \boxed{15.}$$

## 2.11

```

int iResult, num1 = 25, num2 = 40, num3 = 17, num4 = 5;
double fResult, val1 = 17.0, val2 = 12.78;

```

Given the above declarations, we're tasked with determining the result of several statements.

- a. `iResult = num1 / num4` will store the integer 5 in `iResult`.
- b. `fResult = num1 / num4` will store the double 5.0 in `fResult`.
- c. `iResult = num3 / num4` will store the integer 3 in `iResult`.
- d. `fResult = num3 / num4` will store the double 3.0 in `fResult`.
- e. `fResult = val1 / num4` will store the double 3.4 in `fResult`.
- f. `fResult = val1 / val2` will store the double 1.33... in `fResult`.
- g. `iResult = num1 / num2` will store the integer 0 in `iResult`.
- h. `fResult = (double)num1 / num2` will store the double 0.625 in `fResult`.
- i. `fResult = num1 / (double)num2` will store the double 0.625 in `fResult`.
- j. `fResult = (double)(num1 / num2)` will store the double 0.0 in `fResult`.
- k. `iResult = (int)(val1 / num4)` will store the integer 3 in `iResult`.
- l. `fResult = (int)(val1 / num4)` will store the double 3.0 in `fResult`.
- m. `fResult = (int)((double)num1 / num2)` will store the double 0.0 in `fResult`.
- n. `iResult = num3 % num4` will store the integer 2 in `iResult`.
- o. `iResult = num2 % num3` will store the integer 6 in `iResult`.
- p. `iResult = num3 % num2` will store the integer 17 in `iResult`.
- q. `iResult = num2 % num4` will store the integer 0 in `iResult`.

## 2.12

We're tasked with stating the order in which the following expressions will be evaluated by providing a number beneath each operator.

a.  $a - b - c - d$   
1    2    3

b.  $a - b + c - d$   
1    2    3

c.  $a + b / c / d$   
3    1    2

d.  $a + b / c * d$   
3    1    2

## Section 2.2

## Section 2.3