

Midterm Practice

15.11.2021

Java Expressions

Compute the value of each of the following Java expressions. If an error occurs during any of these evaluations, write “Error” on that line and explain briefly why the error occurs.

Java Expression	Value
<code>5.0 / 4 - 4 / 5</code>	1.25 (Starting from left, and first division)
<code>7 < 9 - 5 && 3 % 0 == 3</code>	False (Since the first operand false it doesn't look for the second one)
<code>"E" - "A"</code>	Error: Can't subtract strings
<code>"2" + 1 + 0.009</code>	210.009 (Starting with string)
<code>"1" + (2 + 2 * 3) + 1</code>	181 (First parentheses than string concatenation)
<code>(int) 0.2 / 0.1</code>	0.0 (First casting then division)
<code>int i = (int) Math.random();</code>	0 (Random method gives number between 0 and 1 excluding 1, when we cast it the fraction part is truncated)

Output

- Assume that the method findNumber has been defined as given below. What will be the output?

```
public class OutputTest {  
    public static void main(String[] args) {  
        System.out.println(findNumber(1729));  
    }  
    public static int findNumber(int n) {  
        while (n >= 10) {  
            int k = 0;  
            while (n > 0) {  
                k += n % 10;  
                n /= 10;  
            }  
            n = k;  
        }  
        return n;  
    }  
}
```

n	k
1729	0
172	9
17	11
1	18
0	19
19	0
1	9
0	10
10	0
1	0
0	1
1	

Output is:
1

Output

- Assume that the method findNumber has been defined as given below. What will be the output for findNumber(-6)?

```
public static void FindNumber(int n) {  
    System.out.print(n + " ");  
    if (n > 10) {  
        n = n / 2;  
    } else if (n < 10) {  
        n = n * 2;  
    }  
    if (n % 2 == 1) {  
        n++;  
    } else {  
        n--;  
    }  
    System.out.println(n);  
}
```

Output is:
-6 -13

Output

```
public static void main(String[] args) {  
    String result = "";  
    int x = 7, y = 8;  
    if (x == 3) { result += "1"; }  
    else if (x > 9) { result += "2"; }  
    else if (y < 9) { result += "3"; }  
    else if (x == 7) { result += "4"; }  
    else { result += "5"; }  
    System.out.println (result);  
}
```

Output is:

3

Find the Errors

```
public void main (String[] args) {  
    String X1="me";  
    String X2="me";  
    int x;  
    if(X1 == X2) {  
        x = x + 1;  
    }  
    System.out.println("X = " + x);  
}
```

1. static should be added to the main method's header.
2. x should be initialized
3. Use equals() method instead of ==

Formating

CONVERSIONS BETWEEN CELSIUS AND FAHRENHEIT:

Write a class that contains the following two methods:

```
/** Convert from Celsius to Fahrenheit */  
public static double celsiusToFahrenheit(double celsius)  
/** Convert from Fahrenheit to Celsius */  
public static double fahrenheitToCelsius(double fahrenheit)
```

The formula for the conversion is:

$\text{fahrenheit} = (9.0 / 5) * \text{celsius} + 32$

$\text{celsius} = (5.0 / 9) * (\text{fahrenheit} - 32)$

Write a test program that invokes these methods to display the following tables:

Celsius	Fahrenheit		Fahrenheit	Celsius
40.0	104.0		120.0	48.89
39.0	102.2		110.0	43.33
...				
32.0	89.6		40.0	4.44
31.0	87.8		30.0	-1.11

Formating

```
public class Exercise_1{
    public static void main(String[] args) {
        System.out.println("\nCelcius      Fahrenheit      |      Fahrenheit      Celcius");
        System.out.println("-----");
        double c=40.0, f=120.0;
        double fahrenheit;
        double celcius;
        while (c>=31.0 && f>=30.0){
            fahrenheit=celsiusToFahrenheit(c);
            celcius=fahrenheitToCelsius(f);
            System.out.printf("%-11.2f%-15.2f" + "|" + "%-15.2f%-11.2f" + "\n",c,fahrenheit,f,celcius);
            c-=1.0;
            f-=10.0;
        }
    }
    public static double celsiusToFahrenheit(double celcius){
        double fahrenheit = ( 9.0 / 5 ) * celcius + 32;
        return fahrenheit;
    }
    public static double fahrenheitToCelsius(double fahrenheit){
        double celcius = (5.0 / 9) * (fahrenheit - 32);
        return celcius;
    }
}
```


Nested Loops

Use nested loops to print the following pattern:

Hint: One of your loop counters should have a range dependent on the other counter's current value.

```
A*
B*A*
C*B*A*
D*C*B*A*
E*D*C*B*A*
F*E*D*C*B*A*
```

Nested Loops

```
public class Test {  
    public static void main(String[] args) {  
  
        for (int i = 65; i <= 70; i++) {  
            for (int j = i; j >= 65; j--) {  
                System.out.print((char) j + "*");  
            }  
            System.out.println();  
        }  
  
    } // end main()  
} // end class
```

Programming

Write a static method named `cheerleader` that accepts two integer parameters *lines* and *cheers* and prints a series of "cheer" lines at increasing levels of indentation. The first parameter represents the number of lines of output to print, and the second represents the number of "cheers" per line. For example, the call of `cheerleader(2, 4)` means that you should print 2 lines of output, each containing 4 "cheers." A "cheer" is an occurrence of the word "Go" in the output. Neighboring cheers are separated by the word "Team", so 1 cheer is printed as "Go", 2 cheers as "Go Team Go", 3 cheers are printed as "Go Team Go Team Go", and so on.

The lines you print should be displayed at increasing levels of indentation. The first line displayed should have no indentation, but each following line should be intended by 3 spaces more than the one before it. In other words, the 2nd line of output should be indented by 3 spaces, the 3rd line by 6 spaces, and so on.

You may assume that both parameters passed your method will have values of at least 1.

The following calls demonstrate your method's behavior. Your method should match this output format exactly:

Call	<code>cheerleader(2, 1);</code>	<code>cheerleader(4, 3);</code>	<code>cheerleader(2, 4);</code>
Output	Go Go	Go Team Go Team Go Go Team Go Team Go Go Team Go Team Go Go Team Go Team Go	Go Team Go Team Go Team Go Go Team Go Team Go Team Go

Programming

```
public static void cheerleader(int lines, int cheers) {  
    for (int line = 0; line < lines; line++) {  
        for (int space = 1; space <= line * 3; space++) {  
            System.out.print(" ");  
        }  
  
        System.out.print("Go");  
        for (int cheer = 2; cheer <= cheers; cheer++) {  
            System.out.print(" Team Go");  
        }  
        System.out.println();  
    }  
}
```