Simple Output with Java System Class

System is a pre-defined class in Java which a programmer can use to perform multiple tasks. In this material, we will learn how to use System class to display an output using an object of an output class called "out". The "out" object is created based on two predefined classes, **PrintStream** and **OutputStream**. The "out" object can be used to perform multiple tasks. In this section, we will only cover three different tasks to display information as an output.

1. **System.out.print()**
   The "out" object can call "print()" method to display information with out skipping to a new line.
2. **System.out.println()**
   The "out" object can call "println()", print line, method to display information with a new line "\n" automatically inserted at the end of information.
3. **System.out.printf()**
   The "out" object can call "printf()", print function, method to display information with the ability to format data.

PrintExample.java

```
1  public class PrintExample {
2    public static void main(String [] args) {
3      System.out.print("John Doe\n");
4      System.out.println("1234");
5      System.out.print("IT 130\n");
6      System.out.println(1234);
7      System.out.println("1234"+5);
8      System.out.println(1234+5);
9
10     System.out.println(4-3);
11     System.out.println(4*3);
12     System.out.println(1/3);
13     System.out.println(1/3.0);
14
15     System.out.printf("John Doe");
16     System.out.printf("\n%d", 1234);
17     System.out.printf("\n%f", 1/3.0);
18     System.out.printf("\n%.2f", 1/3.0);
19
20     System.out.print("\nJohn Doe " + 1234 + " " + (1234+5) + " " + (1/3.0));
21     System.out.println("\nJohn Doe " + 1234 + " " + (1234+5) + " " + (1/3.0));
22     System.out.printf("John Doe %d %d %f", 1234,  1234+5, 1/3.0);
23     System.out.printf("\nJohn Doe %d %d %.2f", 1234, 1234+5, 1/3.0);
24     System.out.printf("\n%s %d %d %.2f", "John Doe", 1234, 1234+5, 1/3.0);
25   }
26 }
```

Output:

```
> run PrintExample
John Doe
1234
IT 130
1234
12345
1239
1
12
0
0.3333333333333333
John Doe
1234
0.333333
0.33
John Doe 1234 1239 0.3333333333333333
John Doe 1234 1239 0.3333333333333333
John Doe 1234 1239 0.333333
John Doe 1234 1239 0.33
John Doe 1234 1239 0.33>
```

Explanation of each line with its output

Line 3: Using print() method to display text, John Doe, with an **escape character**, **\n**, that inserts a newline in the text at this point.

Output:

```
John Doe
```

Line 4: Using println() method to display text, 1234. The **method println() automatically inserts a newline** at the end of information displayed.

Output:

```
1234
```

Line 5: Using print() method to display text, IT 130, with the newline escape character at the end. Notice that "IT 130" is automatically displayed on the newline because it follows println() on line 3.

Output:

```
IT 130
```

Line 6: Using println() method to display an integer, 1234. This is different that 1234 on line 3 because since it is an integer, it doesn't need to be in the quotations.

Output:

```
1234
```

Line 7: Using println() method to include two pieces of information, text “1234” and integer 5. The **symbol + appends integer 5 to text 1234** which results in 12345 being displayed.
Output:

```
12345
```

Line 8: Using println() method to include two pieces of information, integer 1234 and integer 5. Since the two pieces of information are both integer, the **symbol + adds integer 5 to integer 1234** which results in 1239 being displayed.
Output:

```
1239
```

Line 10: Using println() method to include two pieces of information, integer 4 and integer 3. Since the two pieces of information are both integer, the **symbol - subtracts** integer 3 from integer 4 which results in 1 being displayed.
Output:

```
1
```

Line 11: Using println() method to include two pieces of information, integer 4 and integer 3. Since the two pieces of information are both integer, the **symbol * multiplies** integer 4 by integer 3 which results in 12 being displayed.
Output:

```
12
```

Line 12: Using println() method to include two pieces of information, integer 1 and integer 3. Since the two pieces of information are both integer, the **symbol / divides** integer 1 by integer 3 which results in 0 being displayed because integer in Java cannot hold any fraction.
Output:

```
0
```

Line 13: Using println() method to include two pieces of information, integer 1 and real number 3.0. Since the two pieces of information are both numbers, the symbol / divides integer 1 by integer 3.0 which results in 0.3333333… being displayed. This is different than 1 / 3 because the denominator, 3.0, indicates that the results of this division should include accurate fractions.
Output:

```
0.3333333333333333
```

Line 15: Using printf() method to display text, “John Doe”. Notice that “John Doe” is automatically displayed on the newline because it follows println() on line 13.
Output:

```
John Doe
```

Line 16: Using printf() method to display an integer, 1234, with an **escape character**, **\n**, that inserts a newline in the text at this point. The method printf() requires a **placeholder** to display an integer and a real number.

- **%d** is used inside the quotations to state that this location is **reserved for an integer** that follows the comma.

Output:

```
1234
```

Line 17: Using printf() method to display a result of division, 1/3.0, with an **escape character**, **\n**, that inserts a newline in the text at this point.

- **%f** is used inside the quotations to state that this location is **reserved for a real number** that follows the comma.

Output:

```
0.333333
```

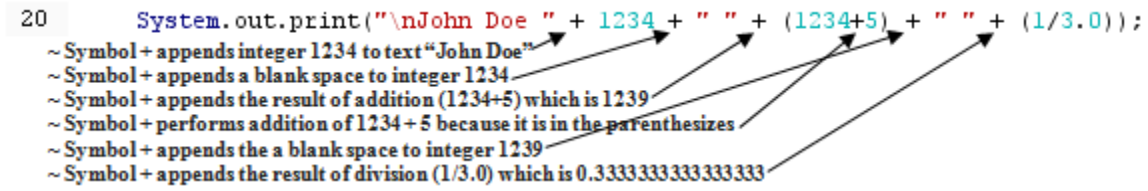
Line 18: Using printf() method to display a result of division, 1/3.0, with an **escape character**, **\n**, that inserts a newline in the text at this point.

- **%.2f** is used inside the quotations to state that this location is reserved for a real number with 2 decimal digits of the fraction that follows the comma.
- The **dot and integer** included in %f indicate how many decimal digit the output should display. For example, %.1f, %.2f, %.3f, i.e.

Output:

```
0.33
```

Line 20: Using print() method to display multiple piece of information by using + symbols to concatenate each one.

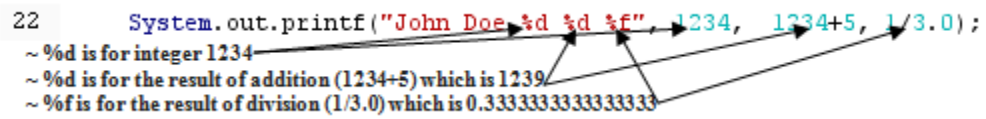


Here is the output of line 20.

```
John Doe 1234 1239 0.3333333333333333
```

Line 21: Producing the same output as line 20 except it is using println().

Line 22: Using printf() method to display multiple piece of information by using multiple placeholders to concatenate each one. When including multiple placeholders, you must match them with the information included in the correct order separated by **comma**.



Here is the output of line 22.

```
John Doe 1234 1239 0.333333
```

Line 23: Producing the output as line 22 except using %.2f as a placeholder to display only two decimal digits 0.33 instead of 0.333333.

```
23      System.out.printf("\nJohn Doe %d %d %.2f", 1234, 1234+5, 1/3.0);
```

Here is the output of line 23

```
John Doe 1234 1239 0.33
```

Line 24: Producing the same output as line 23 except using %s as a placeholder to reserve for a text instead of having "John Doe" itself with the other placeholders. In this case, it will be replaced by "John Doe".

```
24      System.out.printf("\n%s %d %d %.2f", "John Doe", 1234, 1234+5, 1/3.0);
```

~ %s is for integer "John Doe"

NOTE: Placeholders to use for System.out.printf():

Data Type	Placeholder
int	%d
double	%f
String	%s