Unit 1: Primitive Types

Topic 1: println vs. print

Submit in Google Classroom when complete

Name:	

Review slides 23-27 for a summary of today's demo, then **X** here when done! →	

1. Make a mental prediction about how the output of the two code segments below will be different. Then, run each of the two code segments by copying/pasting each into the main method's body. Study the output to find out if and how `System.out.println` and `System.out.print` are different!

Code segment 1

```
System.out.println("Look");
System.out.println("at me!");
System.out.println("Hi!");
```

Code segment 2

```
System.out.print("Look");
System.out.print("at me!");
System.out.print("Hi!");
```

In your own words, how are `System.out.println` and `System.out.print` different?	

2. Consider this code segment:

```
System.out.print("Look");
System.out.println("at me!");
System.out.println("Hi!");
```

Which displays:

```
Lookat me!
Hi!
```

Determine ***three*** different ways you could change the code so that it instead displays

```
Look at me!
Hi!
```

(Kaufman can think of four!)
Be sure to **test** each of your strategies by copying/pasting into the `main` method.

	Copy/paste the updated code segment for each of your solutions:
Solution 1:	
Solution 2:	
Solution 3:	*Need a hint?*
Challenge! **Can you find a fourth possible solution?** (try before peeking)	*some possible solutions*
Did you find a fourth solution *without* peeking at the solutions? ☐	

3. Consider this code segment:

```
System.out.print(Look );
System.out.println(at me!);
System.out.println(Hi!);
```

Predict (using just your eyes and your brains)! Do you think this code will ***compile***? In other words, there would be **no red squiggly lines** if you typed it into Replit. If you *don't* think it would compile, why not? If you think it *will* compile, what will it display when it is executed?	**Write your prediction here:**

4. Now test your prediction by running the code segment.

Was your prediction correct? If not, *why* not, and what did you learn?	

5. Consider this code segment:

```
System.out.println("AP CSA");                    // Line 1
System.out.println("String literal");            // Line 2
System.out.println("A4687BC$");                  // Line 3
System.out.println("* * BOOM! * *");             // Line 4
System.out.println("1 + 2");                     // Line 5
System.out.println("System.out.println");        // Line 6
System.out.println("baad speling");              // Line 7
```

Predict (using just your eyes and your brains)! Do you think this code will compile? If *not*, on which lines(s) are there **syntax/compiler errors** that prevent it from compiling? If it *will* compile, what will it display when it is executed?	**Write your prediction here:**
6. Now copy/paste the code into Replit -- ***don't*** click the run button yet.	
After you've pasted the code, how can you tell that that code *does* compile (i.e. there are no syntax errors)?	
7. Lastly, test your prediction from above by running the program. See if it matches your prediction!	
Any surprises? What can you conclude about printing anything as a string literal (i.e. between two quote marks: " ")?	

check answers

7. Review slides 28 through 35 for a summary on comments and print vs. println, then **X** here when done →	

8. So you just saw that this line of code: `System.out.println("1 + 2");`
prints: 1 + 2
rather than printing: 3

Try executing the line of code *without* using quotes: `System.out.println(1 + 2);`

*Removing the quotes changes 1 + 2 from a **string literal** ("1 + 2") into the **mathematical expression** (1 + 2) which gets simplified **before** being printed!*

What does this tell you about what the `System.out.println()` and `System.out.print()` methods can accept as an **argument** (input to the method)?	*check*

Lab Continues on the Next Page

9. Here is a program that prints out information about Mr. Miller's cats! Pay careful attention to the statements with empty **()**.

```
/* This program prints out info about
   Kaufman's dogs! */

public class Main
{
  public static void main(String[] args)
  {
    // this code displays the info to the screen
    System.out.println("I have 2 dogs.");
    System.out.println();
    System.out.println("Their names are:");
    System.out.print("Holly");
    System.out.println();
    System.out.print("and");
    System.out.println();
    System.out.println("Juniper");
  }
}
```

Predict (using just your eyes and your brains)! What will this code segment display when it is run/executed? Type your prediction *exactly* as you think it would appear.	**Write your prediction here:**

10. Now **test your prediction** by copying/pasting this program into your Main.java file in Replit.

Was your prediction correct? If not, why not, and what did you learn?	

Lab Continues on the Next Page

11. Rewrite the following code as a series of `System.out.println` statements **only** *without using any* `System.out.print` statements:

```
System.out.print("If you will be 18");
System.out.print(" ");
System.out.print("by November 8,");
System.out.println();
System.out.print("Don't ");
System.out.print("forget");
System.out.print(" to register to vote!");
System.out.println();
System.out.println();
System.out.println("THEN VOTE!");
```

TEST your code by copying/pasting into Replit and running it!

Need a hint?

Copy/paste your *rewritten* code here (using println statements only):	*sample solution*

Did you figure out a solution on your own *without* first peeking at the sample solution? ☐	

11. Review slide 36 for a summary of today's class, then **X** here when done! →	

REFLECTION

1. What did you learn about Java syntax rules and compiler errors?
2. What did you learn about the difference between println and print?

Write a reflection below using technical vocabulary from today's lab and slides:

Done!

Submit in Google Classroom:

Turn in

HINTS

Question 2 Hints (jump back):

- You can use more or fewer than three total statements!

- Three of the solutions involve adding a space, for example (“Look “) instead of (“Look”), or creatively using (“ “).

- The fourth solution involves using fewer than three total statements.

Question 11 Hint (jump back):

- You can do this with **four** total `System.out.println` statements.

SOLUTIONS

Question 2 Solutions (jump back):

These are four possible solutions, but you certainly may have found others!

- Possible solution 1:
```
System.out.print("Look ");
System.out.println("at me!");
System.out.println("Hi!");
```
- Possible solution 2:
```
System.out.print("Look");
System.out.println(" at me!");
System.out.println("Hi!");
```
- Possible solution 3:
```
System.out.println("Look at me!");
System.out.println("Hi!");
```
- Another possible solution!
```
System.out.print("Look");
System.out.print(" ");
System.out.println("at me!");
System.out.println("Hi!");
```

Question 11 Solution (jump back):

The code as written outputs the following:

```
If you will be 18 by November 8,
Don’t forget to register to vote!

THEN VOTE!
```

One way to rewrite the code using *only* `println` statements is this:

```
System.out.println("If you will be 18 by November 8,");
System.out.println("Don’t forget to register to vote!");
System.out.println();
System.out.println("THEN VOTE!");
```

Sample answer (back)

Your answer does not have to be exactly this (nor should it be!), but the sample answer below captures the basic idea.

What does this tell you about what the `System.out.println()` and `System.out.print()` methods can accept as an **argument** (input to the method)?	Both methods can accept textual data (strings) or numerical values (two different data types) as the data type of its argument

Sample answers (back)

Your answers do not have to be exactly this (nor should the be!), but the sample answers below capture the basic ideas of each question

Predict (using just your eyes and your brains)!
Do you think this code will compile? If *not*, on which lines(s) are there **syntax/compiler errors** that prevent it from compiling?
 	
If it *will* compile, what will it display when it is executed?
	There are **no** syntax/compiler errors -- all lines of code are properly written and valid! Therefore, it **will** compile.
	
	Here is what gets displayed when executed:
	`AP CSA`
	`String literal`
	`A4687BC$`
	`* * BOOM! * *`
	`1 + 2`
	`System.out.println`
	`baad speling`

6. Now copy/paste the code into Replit -- ***don't*** click the run button yet.	
After you've pasted the code, how can you tell that that code *does* compile (i.e. there are no syntax errors)?	There are **no red squigglies** after you paste it! This means the code has successfully auto-compiled (without syntax errors) and is ready to execute -- see image below:

No red squigglies  means the code **has** successfully **auto-compiled!**

```java
public class Main {
    public static void main(String[] args) {
        System.out.println("AP CSA");                    // Line 1
        System.out.println("String literal");            // Line 2
        System.out.println("A4687BC$");                  // Line 3
        System.out.println("* * BOOM! * *");             // Line 4
        System.out.println("1 + 2");                     // Line 5
        System.out.println("System.out.println");        // Line 6
        System.out.println("baad speling");              // Line 7
    }
}
```

7. Lastly, test your prediction from above by running the program. See if it matches your prediction!	
Any surprises? What can you conclude about printing anything as a string literal (i.e. between two quote marks: " ")?	**Conclusion:** You can print *anything* between double quotes and it will print out *exactly* as typed! This includes symbols, math expressions, and bad spelling!