CS 163/164
Exam 2 Review

Review from first exam

What does this print?

```
String s = “marco polo”;
System.out.println(s.substring(0,3));
```

mar

Print the predefined double variable **d** with 9 decimal place precision (with a new line).

```
System.out.printf(”%.9f\n”, d);
```

Create a Scanner that reads in a word from the keyboard. Store the word in a String variable called **wordsBro**.

```
Scanner s = new Scanner (System.in);
String wordsBro = s.next();
```

Why do you need to have an extra .nextLine() when you are trying to read a full line after calling .next(), .nextInt(), or .nextDouble()?

token processing to line processing.

.nextLine looks for a new line character (’\n’) in a line, so after reading a word, int, double, etc there is still a ‘\n’ character to read, so you must “eat” the rest of the line to read the next line.

Loops

Write a for loop that prints each character in the predefined variable String **s** separated by an ampersand (&) all on the same line.

```
for (int i = 0; i < s.length(); i++)
    System.out.print(s.charAt(i) + “&”);
```

Write a for loop that prints the reverse of the String variable **s**.

```
for (int i = s.length() -1; i >= 0; i--)
        System.out.print(s.charAt(i));
```

Write a while loop that prints the numbers 3 – 9 (inclusive) separated by new lines.

```
int i = 3;
while (i <= 9) {
  System.out.println(i);
  i++;
}
```

Write a do-while loop that will add the sum of all numbers from 1 - 19 (inclusive).

```
int count = 1;
int sum = 0;
do {
  sum += count;
  count++;
} while(count < 20);
```

What does the following code print?

```
for (int l = 0; l > 0; l++)
    System.out.println(l);
```

nothing

Which one of these would not go to the end of String s? (Specify which ones cause errors and which ones are just incorrect (but they compile)?

A. `for (int i = 0; i <= s.length(); i++)`
B. `for (int i = 0; i < s.length()+1; i++)`
C. `for (int i = 0; i < s.length(); i++)`
D. `for (int i = 0; i <= s.length() -1; i++)`
E. `for (int i = 0; i < s.length() -1; i ++)`

A gives an error (goes one more)
B gives an error (goes one more)
C is correct
D is correct
E is incorrect only goes to the second to the last index

Arrays

Declare and allocate a String array called **sArray** to be of size 10.

```
String [] sArray = new String [10];
```

Declare and initialize an int array called **iArray** with the values 1, 2, 3 (in that order).

```
int [] iArray = {1, 2, 3};
```

Given the predefined 1D String array called **stringArray**. Print the length of the array (with a new line).

```
System.out.println(stringArray.length);
```

Declare and allocate a 4x7 2D char array called **letters**.

```
char [][] letters = new char [4][7];
```

Print each element of the predefined 2D byte array called **b** (every element should be printed on the same line, with a new line at the very end)

```
for (int i = 0; i < b.length; i++)
  for (int j = 0; j < b[i].length; j++)
    System.out.print(b[i][j]);
System.out.println();
```

Declare and assign a 3x3 2D double array, called **doubleTable**, with all of the values assigned to 2.0.

```
double [] [] doubleTable = new double [3][3];
for (int i = 0; i < doubleTable.length; i++)
    for (int j = 0; j < doubleTable[i].length; j++)
        doubleTable[i][j] = 2.0;
```

OR

```
double [] [] doubleTable = {{2.0, 2.0, 2.0},
                            {2.0, 2.0, 2.0},
                            {2.0, 2.0, 2.0}};
// spacing doesn't change anything
```

What does the following code print?

```
1  public class Practice {
2    public static void main (String [] args) {
3      int [][] array_name = new int [4][7];
4      int num = 1;
5      for (int row = 0; row < array_name.length; row++)
6        for (int col = 0; col < array_name[row].length; col++)
7          array_name[row][col] = num++;
8      for (int i = 0; i < array_name.length; i++) {
9        for (int j = 0; j < array_name[i].length; j++)
10         System.out.printf("%s ", array_name[i][j]);
11       System.out.println();
12     }
13   }
14 }
```

prints the February calendar

1 2 3 4 5 6 7
8 9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28

What does this code print?
Note: ASCII value of ‘a’ is 97 and ‘b’ is 98.

```
1  import java.util.Arrays;
2
3  public class Practice {
4    public static void Foo (int [] array, char a, char b){
5      for (int i = 0; i < array.length/2; i++)
6        array[i] = a;
7      for (int i = array.length/2; i < array.length; i++)
8        array[i] = b;
9    }
10
11   public static void main (String [] args) {
12     int [] iArray = new int [10];
13     Foo(iArray, 'a', 'b');
14     System.out.println(Arrays.toString(iArray));
15   }
16 }
```

[97, 97, 97, 97, 97, 98, 98, 98, 98, 98]

Methods

1. What is the return value of this method?
2. What does this method do?
3. How would I call this method?

```
1 public class Practice {
2    public static char Foo (String s) {
3       return s.charAt(s.length()-1);
4    }
5 }
```

1. char
2. returns the last character of s
3. Foo(”someString”); or
   Foo(someStringVariable);

1. What is the return value of this method?
2. How would I print the result of this method (in the main method)?

```
1  public class Practice {
2    public static boolean fullOCleverNames () {
3     int i = 32;
4     double d = 32.3;
5     if (i == Math.floor(d))
6       return true;
7     else
8      return false;
9    }
10 }
```

1. boolean
2. System.out.println(fullOCleverNames());

Create a public static method called **caster**, that returns a double, it takes an int as a parameter. Return the double, caused from typecasting the parameter.

```
public static double caster (int i) {
        return (double)i;
}
```

Create a public static method called **printMe**, that returns nothing and takes a float as a parameter. Print the float with 8 decimal points with a new line.

```
public static void printMe (float f0) {
    System.out.printf("%.8f\n", f0);
}
```

Create a public static method stub (no code inside) called **practice**, that returns a char and takes an int and a String as a parameter.

```
public static char practice (int i, String s);

// you can have any parameter variable names
```

Create a public static method stub (no code inside) called **practice1**, that returns nothing and has no parameters.

```
public static void practice1 ();
```

What does the following code print?

```
1  import java.util.Arrays;
2
3  public class Practice {
4    public static void multiplier (double d, int [] array) {
5      for (int i = 0; i< array.length; i++)
6        array[i] *= d;
7    }
8    public static void main (String [] args){
9      int [] array = {1, 2, 3, 4, 5};
10     multiplier(3, array);
11     System.out.println(Arrays.toString(array));
12   }
13 }
```

[3, 6, 9, 12, 15]