

Java...



CS 163/164
Exam 2 Review

**Please, what does `[arg1, [arg2]`
mean....**

Review from first exam

Name the 8 primitive type variables

int, double, float, short, long, char,
boolean, byte

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();
```

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));
```

Declare a Scanner that reads from the keyboard. Write a do-while loop that asks the user “Are we there yet?”, give the user instructions (for example “yes will stop this loop”), store their response into a String variable, and stop the loop when the word is “yes”. NOTE: you can assume that the user will only enter a one word response.

```
Scanner keys = new Scanner (System.in);
String word = "";
do {
    System.out.println("Are we there yet?");
    System.out.println("Enter yes to stop");
    word = keys.next();
    word = word.toLowerCase();
} while (!word.equals ("yes")); // could also use
                                // equalsIgnoreCase
```

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.

Review for exam 2: 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 do?

```
int [] [] array_name = new int [4][7];
int i = 1;
for (int row = 0; row < array_name.length; row++)
    for (int col = 0; col < array_name[row].length; col++){
        array_name[row][col] = i;
        i++;
    }
```

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 do?

```
public void Foo (int [] array, char a, char b){  
    for (int i = 0; i < array.length/2; i++)  
        array[i] = a;  
    for (int i = array.length/2; i < array.length; i++)  
        array[i] = b;  
}
```


Makes the first half of array the ascii value of the a variable and the second half of the array to be the ascii value from the b variable.

Review for exam 2: methods

```
public class StudySession {  
    public static char Foo (String s) {  
        return s.charAt(s.length()-1);  
    }  
}
```

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

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

```
public class StudySession {  
    public void Foo1 (char c, int i) {  
        System.out.println(c + i);  
    }  
}
```

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

1. void (doesn't return anything)
2. adds the ascii value of c with i (prints a number not a character)
3.

```
public static void main (String [] args) {  
    StudySession SS = new StudySession();  
    SS.Foo1(some char, some number);  
}
```

```
public class StudySession {  
    public static boolean fullOCleverNames () {  
        int i = 32;  
        double d = 32.3;  
        if (i == Math.floor(d))  
            return true;  
        else  
            return false;  
    }  
}
```

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

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 non-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 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 ();
```



```
public class CoolName {  
    public static void multiplier (double d, int [] array) {  
        for (int i = 0; i < array.length; i++)  
            array[i] *= d;  
    }  
    public static void main (String [] args){  
        int [] array = {1, 2, 3, 4, 5};  
        multiplier(3, array);  
        System.out.println(Arrays.toString(array));  
    }  
}
```

What is printed?

[3, 6, 9, 12, 15]

Review for exam 2: Objects

Make a StudentData object called student1 with id: 123, first name: James, last name: Bond

```
public class StudentData {  
    private int id;  
    private String firstName, lastName;  
    public StudentData (int num, String first, String last) {  
        id = num;  
        firstName = first;  
        lastName = last;  
    }  
}
```

```
StudentData student1 = new StudentData (123, “James”,  
“Bond”);
```

Declare and allocate an array of StudentData objects called “CS163” to have a size of 3.

```
public class StudentData {  
    private int id;  
    private String firstName, lastName;  
    public StudentData (int num, String first, String last) {  
        id = num;  
        firstName = first;  
        lastName = last;  
    }  
}
```

```
StudentData [] CS163 = new StudentData [3];
```

Use the following information to create three StudentData objects and store into the CS163 array from Part A.

- 1 Rob Drobs
- 2 Bob Crobs
- 3 Steve Jobs

```
public class StudentData {
    private int id;
    private String firstName, lastName;
    public StudentData (int num, String first, String last) {
        id = num;
        firstName = first;
        lastName = last;
    }
}
```



```
CS163[0] = new StudentData (1, "Rob", "Drobs");  
CS163[1] = new StudentData(2, "Bob", "Crobs");  
CS163[2] = new StudentData(3, "Steve", "Jobs");
```

OR

```
StudentData sd1 = new StudentData(1, "Rob", "Drobs");  
CS163[0] = sd1;  
StudentData sd2 = new StudentData(2, "Bob", "Crobs");  
CS163[1] = sd2;  
StudentData sd3 = new StudentData(3, "Steve", "Jobs");  
CS163[2] = sd3;
```

Write a toString method such that when an object is printed the format is:
id, lastName, firstName

```
public class StudentData {  
    private int id;  
    private String firstName, lastName;  
    public StudentData (int num, String first, String last) {  
        id = num;  
        firstName = first;  
        lastName = last;  
    }  
}
```

```
@Override
public String toString () {
    return id + ", " + lastName + ", " + firstName;
}
```

OR

```
@Override
public String toString () {
    return this.id + ", " + this.lastName + ", " + this.firstName;
}
```

Note: Override is optional, but suggested

When do you need to use the keyword “this”?

When you are accessing class variables that have the same name as a local variable.

E.g. if (this.name == otherObject.name)

OR

```
public class Student {  
    private String name;  
    public Student (String name) {  
        this.name = name;  
    }  
}
```