

# Griffin's OH 2/4/22

COMS W1004 Introduction to Computer  
Science and Programming in Java





# Announcements and Updates

- **Quiz 1** Grades have been released, please go check your score if you have not already on Courseworks
- **Problem Set 1** was due on **1/31/2022** if you have still not submitted yours yet please do so ASAP as we have begun grading and want to get them back to you within the next few days
- **Programming Problem Set 1** is due on **2/7/2022** please begin working on yours if you have not already done so.
- **Quiz 2** is on **2/24/2022** please begin to prepare for it now so you are not anxious when the day comes
- Starting this **Sunday (2/6/2022)** we will have a room reserved in **Math 207** from **1-5pm** that serve as extended office hours for multiple TAs and a place to work on the upcoming problem set. Please use your best judgment to determine whether or not these sessions will be helpful to you



# Topics for the Week

1. Review the Quiz on 2/1/2022
2. Review Elementary Coding Components
3. New: Boolean expressions and Operators
4. New: Arithmetic Operators and Order of Precedence
5. General Questions About Programming Project



# Thoughts on the Quiz

I know several of you are disappointed with the score you received on your quiz this past Tuesday. I just wanted to let you know that it is okay and that the first quiz for any class is usually the worst one since you are still trying to understand the study for the course. This happens to everyone throughout their academic career and especially in Computer Science since the learning curve is a bit steep. I myself was absolutely terrible when I started programming.

Please remember though: ***\*FAILURE IS A PART OF THE PROCESS\****



## Question 1 - Insertion Sort

```
for(j=2; j ≤ n; j++){
    temp = A[j]
    i = j - 1
    while(i > 0 and A[i] > temp){
        A[i+1]=A[i]
        i=i-1
    }
    A[i+1] = temp
}
```

Consider the list of numbers  $A=(3,1,4,2,8,0)$ . Recall the insertion-sort algorithm from class and suppose it is used to sort this list in ascending order. What will the list look like after 3 passes (iterations of the outer loop)?

Let's start by reminding ourselves of the insertion sort algorithm:



## Question 1 - Insertion Sort

Using the algorithm from the previous slide we know that after each iteration of the outer loop the algorithm will look as follows:

1. 1 3 4 2 8 0
2. 1 3 4 2 8 0
3. 1 2 3 4 8 0
4. 1 2 3 4 8 0
5. 0 1 2 3 4 8

So the answer to the question would be the set of numbers on line 3



## Question 2 - Selection Sort

```
for(int i = 1; i < n; i++){  
    int locmin = i;  
    for(int j = i+1; j < n+1; j++){  
        if(A[locmin] > A[j]){  
            locmin = j;  
        }  
    }  
    exchange(A[locmin], A[i]);  
}
```

Consider the list of numbers  $A=(6,1,2,5,4,10)$ . Recall the selection-sort algorithm from class and suppose it is used to sort this list in ascending order. What will the list look like after 4 passes (iterations of the outer loop)?

Let's remind ourselves of the Selection Sort Algorithm:



## Question 2 - Selection Sort

Using the algorithm from the previous slide we know that after each iteration of the outer loop the algorithm will look as follows:

1. 6 1 2 5 4 10
2. 4 1 2 5 6 10
3. 4 1 2 5 6 10
4. 2 1 4 5 6 10
5. 1 2 4 5 6 10

So the answer to our specific question would be the set of numbers on line 4





## Question 3 - Insertion Sort Comparisons

How many comparisons does the insertion-sort algorithm seen in class take to sort the list

$A=(3,1,4,2,8,0)$ ?

The key to this question is look at the list and think about how out of order the numbers are, if they are in descending order then it will take the most comparisons but if it is already in descending order then it will only have to make 1 comparison each iteration so just write down how many comparisons you do on paper and count them up.



## Question 4 - Binary Search Comparisons

Consider the sorted list (1,2,4,5,6,10,13,16). How many comparisons will binary search perform on this list when searching for the 2?

You do not need to even know the whole algorithm for binary search, just remember it cuts the list in two with each iteration.

First comparison is 2 and 5 since  $2 < 5$  the right side of the list will be chopped off

Second comparison is 2 and 2 since  $2 == 2$  we have found the match and can quit



## Question 5 - Binary vs Linear

Recall the difference between searching a sorted list with binary search and searching an unsorted list with linear (sequential) search. If a list has 17 elements in it, what is the minimum number of searches necessary to make first sorting the list and then using binary search better than just using linear search in terms of the total number of comparisons over all searches? Assume selection-sort is being used to sort the list and be sure to use the formulas exact number of comparisons, not their order of growth.

This is just like Question 33 from the first Problem Set!

$$n(n-1)/2 + p \log n = pn$$

$$p = n(n-1)/2 / (n - \log n)$$

Then just plug in your value for  $n$  and you get  $p$ !



# Review: Elementary Coding Components

1. Class definition: All java code is encapsulated within a class, the classname must match the filename
2. Main method: To run a Java file you must have a main method present within your file to run your code, you'll still be able to compile without one but to run you must have it
3. Data-types: Java is a strongly typed programming language everytime you declare a variable you must give it a type, whether a primitive data type or an Object data type



# New: Boolean Expressions and Operators

In Java you will learn more and more about decision making and how to direct the flow of your program. You will learn how to do this using conditional statements, within conditional statements are boolean expressions. Boolean Expressions are expressions that evaluate to true or false and depending on how you structure your conditional statements change the flow of your program. Here is a simple boolean expression that you are already familiar with

`A[i] == someValue` or `A[i] != someValue`

You are used to seeing these in conditional statements already but there are so many operators, what do they all mean?



# New: Boolean Expressions and Operators

Operator	Meaning
==	Equal to
!=	Not equal to
	Or
&&	And
>, <, >=, <=	Greater than, less than, greater than or equal to and less than or equal to
!isRaining	Not (Ex !true = false)



# New: Arithmetic Operators and Order of Precedence

Just as there are boolean operators, there are ones to perform mathematical operations as well.

Operators	Meaning
+, -, ++, --	Add two values, subtract two values, increment by 1, decrement by 1
*, /, %	Multiply two values, divide two values, gets the remainder of division
+= -=	Same as saying $V = V + S$ or $V = V - S$
*= /=	Same as saying $V = V * S$ or $V = V / S$



# New: Arithmetic Operators and Order of Precedence

Operators	Precedence
!, ++, − (unary operators)	First (Highest)
*, /, %	Second
+ −	Third
< <= >= >	Fourth
== !=	Fifth
&&	Sixth
	Seventh
= *= += -= /= %=	Lowest