Java
John Marchand – jhmarchand@gmail.com
Team 4381 – Twisted Devils
https://github.com/jhmarchand/java4381

Week 4

Trying to save some space, I will only be showing snippets of code. Full code is on github week4.

Tic Tac Toe

Last week, I asked you to print out a tic tac toe board using a for loop and an array. I have decided that was a bad example, and we'll try again this week. First lets look at my solution.

As I was writing my code, I really decided this was the best way to print out the board.

```
System.out.println(topRow[0] + "_|_" + topRow[1] + "_|_" + topRow[2]);
System.out.println(middleRow[0] + "_|_" + middleRow[1] + "_|_" + middleRow[2]);
System.out.println(bottomRow[0] + " | " + bottomRow[1] + " | " + bottomRow[2]);
```

In the following code I check to see if there is winner in the top row. To have a winner, we need to have the same character in every spot on the top row. We can only compare 2 things at a time, so logically this works out to is the first character equal to the second character AND is the second character equal to the third character AND is that character not a space. So we have 3 conditions and they all have to be true for us to consider this whole condition true. In code the AND is represented by && as you can see below.

```
if ( topRow[0] == topRow[1] && topRow[1] == topRow[2] && topRow[0] != ' ' )
{
      weHaveAWinner = true;
      winner = turn;
}
```

If you have two conditions and you want to say If this OR that and you want to execute code if either one is true, you can use || to represent or.

We have learned that while and for loops run until the condition is false. However, there is another trick we can use from time to time. Its called a break statement. Basically, if the code hits a break statement, it goes to the end of the loop immediately. Below shows an example of using the break statement to get out of the loop when we have a winner. Otherwise we would stay in the loop until all 9 spots are filled in.

```
if ( weHaveAWinner )
      break;
```

Functions

Lets consider the following code...

```
java.util.Scanner input = new java.util.Scanner(System.in);
row = input.nextInt();
```

nextInt() is a function. It represents a block of code that you can use over and over. The code runs and this function returns an int.

```
System.out.println("Winner is " + winner);
```

println(String s) is a function that takes a parameter of type String. Its return type is void, which means it doesnt return any information.

It turns out you can write your own functions so that your code is more readable, as well as easier to maintain. Most importantly though, if you are doing the same thing multiple times, you only have to write it once.

I rewrote TicTacToe.java as TicTacToeFun.java using functions. Lets look at my while loop.

```
while ( turnCount++ < 10 && winner == ' ' ) {
	printBoard(topRow, middleRow, bottomRow);

	getMove(input, topRow, middleRow, bottomRow, turn);

	winner = checkForWinner(topRow, middleRow, bottomRow);

	if ( turn == 'X')
		turn = 'O';
	else
		turn = 'X';
}
```

Thats easier to follow all the logic.

You can go to github to see the rest, but for now, lets start with some smaller functions.

Consider the following code

```
public static void main(String[] args) {

	int a = 7;
	int b = 8;

	int sum = add(a,b);

	System.out.println("sum = " + sum);


}

private static int add(int a, int b) {

	int c = 0;

	c = a+ b;

	return c;
}
```

Lets ignore private static part for now. Lets talk about the rest. The function is called add. It has a return type of int. And it takes 2 parameters both of type int. Look at the last line return c says go back to the calling line of code and return the value of c.

Can you write 3 more functions for me? Using add as a template, write a subtract, multiply and divide function.

Arrays Again

Let say Im a teacher and I want to total up all your test scores. If I have 5 test scores, I could write a function that takes 5 ints and add them all up and return the result. Thats great, but what if I give another test? Write another function? Add another variable? It turns out arrays are great for holding a list of items that all belong together. Consider an array I start out with

```
int[] testScores = { 90,95,87, 100, 96};
```

Now remember testScores[0] = 90, and testScores[3] = 100. So if I use that array, I can write 1 function that will total any number of ints. Consider this.

```
private static int getSum( int[] intArray) {
	int sum = 0;

	for ( int i = 0; i < intArray.length; i++ )
		sum = sum + intArray[i];

		return sum;

}
```

notice int.length, that allows this for loop to go through all of the integers in the array, even though at the time I write the function, I dont know how many ints I am totaling up.

To give you a grade, a teacher has to take an average of your test scores. Can you write a function that will give me the average?

Calculator Part 2! SUPER BONUS EXTRA EXTRA

How about you let the user type something like 57+64. Can you parse that and figure out the result? Remember that strings are like a list of char. The trick is to create 2 strings and a char from that. You want the first string to be "57", a char to be '+' and a second string that is "64". From there you can use if statements to determine if you should add, subtract multiply or divide. Here's a few helpful hints.

For a given string you can convert it to a char[] using the following.

```
String inputString = "57+64";
char [] inputCharArray = inputString.toCharArray();
```

Don't forget to use inputCharArray.length!

You can't add 2 strings that are numbers and expect to get the sum. 5 + 6 = 11, but "5" + "6" = "56". So you have to convert them to numbers. The following is a simple way to do that.

```
String numberString = "5";
int a=0;
a = Integer.parseInt(numberString);
```

Good Luck. Next week I think we are ready to discuss classes and objects in more detail.