

Java

John Marchand – jhmarchand@gmail.com

Team 4381 – Twisted Devils

<https://github.com/jhmarchand/java4381>

Week 2

You may go to the git repository listed above to get his pdf as well as source code for each week. I will only have snippets in this pdf, but the full source is available in the github repository.

Today we will discuss the final program from last week, your very own adding machine. Next we will learn integer and string assignments and manipulation. Then we will move to for loops, while loops and if statements.

Adding Machine – AddingMachine.java

Here is the code for the final problem from last week. Get 2 numbers from the user and add them together. Notice there is 2 right ways and 1 wrong way to do this

```
System.out.println("Enter a number");
int firstNumber = scanner.nextInt();

System.out.println("Enter a number");
int secondNumber = scanner.nextInt();

int sum = firstNumber + secondNumber;

// 2 of these are right, do you know which ones?
System.out.println(firstNumber + " + " + secondNumber + " = " + sum);
System.out.println(firstNumber + " + " + secondNumber + " = " + firstNumber + secondNumber);
System.out.println(firstNumber + " + " + secondNumber + " = " + ( firstNumber +
secondNumber) );
```

More Variable Fun

Remember, variables are used to store information for our programs. We can store the users name, or their score, or anything we want. Lets break down a variable declaration.

```
int myNumber = 0;
```

So when we tell the compiler we need a variable, we first tell it the type (int or string), what we want to call it, and the value it should start with.

Once we have some variables, we can add them together

```
myNumber = someNumber + otherNumber;

myNumber = someNumber - 3;

myNumber = someNumber * otherNumber;

myNumber = someNumber / otherNumber;
```

Even Add them to themselves. Lets say we are keeping score at a Gull Lake Blue Devils football team. If we have a variable called blueDevilsScore, we can just increment it by 7 after they score a touchdown. Notice that the same variable is on both sides of the equal signs. So if before this statement, blueDevisScore was 14, after this executed what value would it hold?

```
blueDevilsScore = blueDevilsScore + 7;
```

We can do the same types of assignments with strings! Look at the following examples...

```
String name = "John";
int age = 47;
myString = "My Name is " + name + ", age is " + age;

//adding to itself
myString = "The programmer is ";
myString = myString + name;
```

Strings cant be subtracted though, this will show an error!

```
myString = myString - name;
```

Lets Count to 10

There are many ways to count to 10, some easier than others. Lets go with what we know.

```
System.out.println(1);
System.out.println(2);
System.out.println(3);
System.out.println(4);
System.out.println(5);
System.out.println(6);
System.out.println(7);
System.out.println(8);
System.out.println(9);
System.out.println(10);
```

Easy and you can do that today, lets increment a variable to do the same thing!


```
System.out.println(counter++);
System.out.println(counter++);
System.out.println(counter++);
```

Now that is a pretty clean and easy way to count to 10, but surely we can find a better way? It turns out that many times in your program you will want to do the same thing over and over again. Most programming languages provides loops to make that easy.

A while loop tells the program to do something as long as a condition is true. A condition is usually a comparison of 2 numbers or 2 strings. We will cover conditions more in depth in just a bit. So lets make a while loop that says, while our counter is 10 or less, print it out, and increment.

```
while ( counter < 11 ) {
    System.out.println(counter);
    counter = counter + 1;
}
```

So it loops through there as long as counter has a value less than 11, which is 10 or less! 1 print statement but we get it to print 10 times! How easy would it be to make it count to 100?

Again, the while loop says, while a condition is true, do all the code in between the { }.

Another type of loop is a for loop. So the syntax of a for loop is for (init statement; condition check; incremter). It is very similar to a while loop, but allows you to declare a variable, set a check condition and specify how you want to increment all in one line.

```
for( int i = 1; i < 11; i = i + 1 ) {
    System.out.println(i);
}
```

What happens if the check condition is false in the beginning? Does it even print once? Can you make this count to 10 by only even numbers, how?

If, conditions and a simple guessing game.

Sometimes we only want to execute code if a certain condition is met. Lets look at some comparison operators.

== equal

!= not equal

< less than

> greater than

<= less than or equal to

>= greater than or equal to

We can compare literals, variables or a combination.

```
5 > myNumber  
myNumber == otherNumber
```

Please note that for equality we use == instead of just =, it will give you an error if you try to use a single = in most cases, but still be careful.

So the if statement allows us to test a condition, and if true, it executes the code inside the { }, otherwise it skips over it.

```
if ( blueDevilsScore > lakeViewScore ) {  
    System.out.println("Gull Lake Blue Devils WIN!");  
}  
System.out.println("This Prints no matter who won");
```

You can also leave out the { } it will then only execute the very next statement.

```
if ( blueDevilsScore > lakeViewScore )  
    System.out.println("Gull Lake Blue Devils WIN!");  
System.out.println("This Prints no matter who won");
```

Now we look at a simple guessing game. The point is to guess a number between 0 and 9. The program will tell you how you did. Change the default value of programmers guess to your guess before you run it.

The following line of code puts a random value in the integer variable called answer. It will be a different value every time you run the code.

```
int answer = (int) (Math.random() * 10);
```

Math.Random returns a random number and I do a little conversion on it, for now, lets just accept that it does what it does.

Type up the program on the next page and get it running, then move onto the final programming problem for the week.

```

public class IfGuesser {

    public static void main(String[] args) {

        // Creates a random number from 0 to 9
        int answer = (int) (Math.random() * 10);

        // you could make your guess using the input scanner, but this is just as good for
now.
        int programmersGuess = 7;

        if ( programmersGuess == answer )
            System.out.println("You are right! The answer was " + answer);

        if ( programmersGuess != answer )
            System.out.println("You are wrong! The answer was " + answer);

        if ( programmersGuess > answer )
            System.out.println(programmersGuess + " was too high!");

        if ( programmersGuess < answer )
            System.out.println(programmersGuess + " was too low!");
    }
}

```

Final Program

Write a program that has the user guess a number from the console. If they don't get it right, tell them if they are too high or too low and have them guess again until they get it right.