Methods/Class intro

```java
int getRandomNumber()
{
  return 4;  // chosen by fair dice roll.
  // guaranteed to be random.
}
```
Methods/functions

How to make the person run?

RUN FOREST, RUN!!!

(See: RunForest.java)
Methods/functions

You can also use functions that return boolean types in an if statement or loop.

This is commonly used if you have complex logic as it is normally easier to write a function that have a very complex boolean expression.

(See: FindPrime.java)
(See: sillySwap.java)

Sometimes the value of variables is copied and not given access to the real value

This is similar to moodle, the score you see for grades cannot change the score I give you!
Blocks (inside { }) of code can only see variables from their parent blocks.

You can also make member variables outside of any function, but inside the class (I will often just call these variables inside the class or the class's variable).

(See: MemberVariable.cpp)
Using what we know, we actually cannot swap (change values inside functions that also change values outside the function)

However, this is only really true for the primitive types (the blue ones):

\[
\text{int, boolean, double, char, ...}
\]

All other types are called \text{classes}, such as String, Scanner, .... (capitalized convention)
class

We have actually been using classes since our first program

Everything in Java must be in a class

`public class AddMethod`

(Remember the filename must be the same as the class, so AddMethod.java)
Classes are simply a way of organizing data

(See: Partner.java)
Every time you do not make a primitive type (int, double, boolean, etc.) or String, you must use `new` command.

Basic format (syntax):
```
[class] [identifier] = new [class]();
```

Example:
```
Partner p1 = new Partner();
Scanner in = new Scanner(System.in);
```
Suppose we wanted to keep scores of lab0, lab1, lab2, lab3 and hw1 for 5 students.

Without classes how would I have to do this?

With classes how would I do this? (See: Classes.java)
Debugging

- Test small pieces of code at a time
- Add print statements to see values in loops (and to localize error in general)
- Test code on inputs you know the answer