Formatting

a. Every subsection is indented four spaces. NEVER USE TABS, ONLY USE SPACES.
    i. Example:
```
if(statement) //NEVER DO THIS
{
execute
}

if(statement)//CORRECT FORMATTINg
{
        Execute
}
```
b. All curly braces should be treated as their own statement, and thus, be on their own line
    i. Example:
```
if(statement){ //NO
        Execute
}else if(statement){
        Execute
}

if(statement) //YES
{
        Execute
}
else if(statement)
{
        Execute
}
```
c. Binary operators should have a space on either side.
    i. Example:
```
a=b+c //NO
a = b + c //YES

a=(2*3)+4//NO
a = (2 * 3) + 4 //YES
```
d. Casts, method declarations, and if, while, for, switch etc should be immediately followed by a left parenthesis.
    i. Example:
```
If (statement) //NO
{
        Execute
}

if(statement) //YES
{
        Execute
}

method (parameter); //NO
method(parameter); //YES

(int) number; //NO
(int)number; //YES
```

e. Unary Operators should immediately precede their variable.
    i. Example

```
count ++; //NO
count++; //YES
```

f. Always have fields, then constructors, and then methods when writing classes.
    i. Example

```
class classname
{
        //fields here

        //constructors here

        //methods here
}
```

g. No lines exceeding 80 Characters. Really, if it looks a little too long, break it into multiple lines.

h. Use Parenthesis, even when you don't really need to, to make order of operations more clear. IE: when in doubt, add parenthesis.
    i. Example

```
if(statement && statement || statement)//NO
if((statement && statement) || statement)//YES
```

Identifiers

a. Variable names should always be camel case and be descriptive as to their use. DO NOT specify scope or type as part of a variable name. It violates abstraction principles and wastes typing time.
    i. Example

```
private final String a; //NO
private final String stringA; //NO
private final String privateStringA; //NO
private final String privateStringName; //NO
private final String stringName; //NO
private final String lastname; //NO

private final String name; //YES
private final String lastName; //YES
```

b. Fields should ALWAYS be private, unless you know an outside class needs to get them, in which most cases you should use a method, and ALWAYS be final, unless you ABSOLUTELY KNOW that they will need to be modified.
c. Class names and interfaces should also be camel case. (see a)

Coding Practices

a. Keep return to the end of methods.
b. Initialize variables as close as possible to declaration.
    i. Example:

```
int total;
int num1 = 10;
int num2 = 10;
total = num1 + num2; //NO

int num1 = 10;
int num2 = 10;
int total = num1 + num2; //YES
```

c. Never use unary operators before the variable.
    i. Example:

```
++i; //NO
I++; //YES
```

d. Keep unary operators to their own line.
    i. Example

```
method(x++) //NO
method(x)
x++; //YES
```

Commenting

Use comments. Use them to explain what a method does and stuff. Don't put one after every variable, but, when in doubt, add a comment. It makes the code easier to debug and step through later. And for other people. So use comments. They're helpful.

THESE RULES ARE NOT THE END ALL. THESE RULES SHOULD BE FOLLOWED ONLY WHEN THEY WILL NOT HARM THE CODE OR THE READIBILTY. FUNCTIONALITY MUST COME FIRST.

But make sure your functionality is not making the code unreadable.