

Here are some notes to help simplify what we have learned so far. These notes do not represent everything that you must know for the midterm. For the midterm, you are responsible for all of the material that we have covered in class, lab, prelab, and assignment; except for **if** and **while** statements.

An **expression** is a combination of one or more variables or values that are evaluated to produce a single value.

Assuming a, b and c are numbers, evaluating :

```
5 + 6      produces the number 11
a + 5      produces a number
a++        produces a number
(a + b)/c  produces a number
a % b      produces a number
a == b     produces a Boolean value (true or false)
```

There are 4 basic **statement types**:

1. **Declaration** which declares a variable that references some object.

```
int x;
JFrame y; /* note that y does not refer to a JFrame until you create and
           assign one. */
```

2. **Assignments** which assigns the reference or value on the right hand side of the equals to the variable on the left hand side.

```
x = 5;
y = new JFrame(); /* we use new to create a new object, the value
                  returned is assigned to y */
```

3. **Method calls**

```
y.show(); /* method called with object dot method signature */
System.out.println("test"); /* static method called with class name */
```

4. **return** which return a single value to the method calling statement, for all methods that are not void.

```
return true;
return 8 + y * 2;
return x;
```

More complex combinations - statements which direct execution flow:

if statements permit execution of a body of statements only if the prestated condition is met.

```
if (condition) {
    statements to be performed if condition is true;
}

if (b1) {
    statements to be performed if condition b1 is true;
} else {
    statements to be performed if condition b1 is not true;
}
```

```
if (b1) {
    statements to be performed if condition b1 is true;
} else if (b2) {
    statements to be performed if condition b2 is true;
}
```

while statements execute the body of statements as long as the condition remains satisfied.

```
while (condition) {
    statements to be repeatedly executed.
}
```

Other Observations and questions

== versus equals: What is the difference between == and .equals(object o)?
== can be used with numbers and char because they are primitives. All other non-primitive objects must be semantically compared using the equals method. If you use == to compare two non-primitive objects, then you are verifying whether they refer to the same object in memory, not whether their contents are the same (semantically).

Every class that we write is a **customization of the Object class**.

Given: **double n;**
 (int) n casts n from a **double** to an **int**.

Given: **int x;**
 (double) x casts x from an **int** to a **double**.

How do you change a String to an int and a double?

What operations can be performed on numbers - mathematical expressions?

What operators are used in Boolean expressions?

How do you define a class? a method?

When do you use static?