Computing Basics for use with EJS *(useful for correcting **What Is Wrong?** exercises)*

Variables

Java requires that you alert it to each variable (define variables) so you will see things like the following (notice that every line is followed by a semi-colon):

```
int j;
double x=0.0;
```

`j` is an integer
`x` is a number (equal to 0)

Math Functions

In order to use math functions commonly used in physics, you need to call on a Math library. Some useful ones are

`Math.sin(x)`	$\sin x$ (and similarly for $\cos x$, $\tan x$)
`Math.atan(x)`	$\tan^{-1} x$
`Math.exp(x)`	e^x
`Math.sqrt(x)`	$\sqrt{x}$
`Math.pow(x,3)`	x^3
`Math.max(a,3)`	= a or 3, whichever is greatest
`Math.min(b,-1)`	=b or -1, whichever is smallest

`x=x+1` `(x+=1)` statements:

```
x = x+1.0;
```

the new value of `x` is equal to the old value of `x` plus `1`

```
x+=1.0;
```

short version of the statement above

`If… else` statements:

There are times you want to avoid certain conditions (in the example below, you don't want to divide by zero) and so you use this type of statement for checking:

```
if (r==0.0){
      e=0.0;
}
else {
      e=1/(r*r);
}
```

check `if r=0`
yes, set `e=0`
otherwise
set `e=1/r`^{2}

Arrays and `For` Loops

An array stores a set of values in one variable with labeled (indexed) parts. For example, if you wanted `x` to be the set of the first seven perfect squares {1,4,9,16,25,36,49}, you would want:

```
double[] x=new double[7];
x[0]=1.0;
x[1]=4.0;
x[2]=9.0;
x[3]=16.0;
x[4]=25.0;
x[5]=36.0;
x[6]=49.0;
```

define the array `x` to have 7 elements

Generally, though, you don't put the values in like this, instead, you use a loop:

```
for (int i=0; i<7;i++){
x[i]=(i+1)*(i+1);
}
```

`for` loop: repeats itself for a while (until a condition is met)

So here is a walk-through of what happens as the computer steps through the `for` loop:

```
for (int i=0;i<7;i++){
x[i]=(i+1)*(i+1);
}
```

1st time through: set up an integer `i=0`

```
for (int i=0;i++;i<7){
x[i]=(i+1)*(i+1);
}
```

next time through: increment i by 1, but right now, i=0

```
for (int i=0; i<7;i++){
x[i]=(i+1)*(i+1);
}
```

if i<7 (and it is) continue on

```
for(int i=0; i<7;i++){
x[i]=(i+1)*(i+1);
}
```

x[0]=1*1=1

```
for (int i=0;i<7;i++){
x[i]=(i+1)*(i+1);
}
```

loop back and increment by one so that i=1 and check that i<7

```
for (int i=0;i<7;i++){
x[i]=(i+1)*(i+1);
}
```

x[1]=(1+1)*(1+1)=2*2=4

```
for (int i=0;i<7;i++){
x[i]=(i+1)*(i+1);
}
```

loop back and increment by one so that i=2 and check that i<7

and so forth (skipping a few steps)… to get to

x[6]=(6+1)*(6+1)=7*7=49

```
for (int i=0;i<7;i++){
x[i]=(i+1)*(i+1);
}
```

loop back and increment by one so that i=7 and check that i<7. It doesn't, so leave loop and go to next part of program.