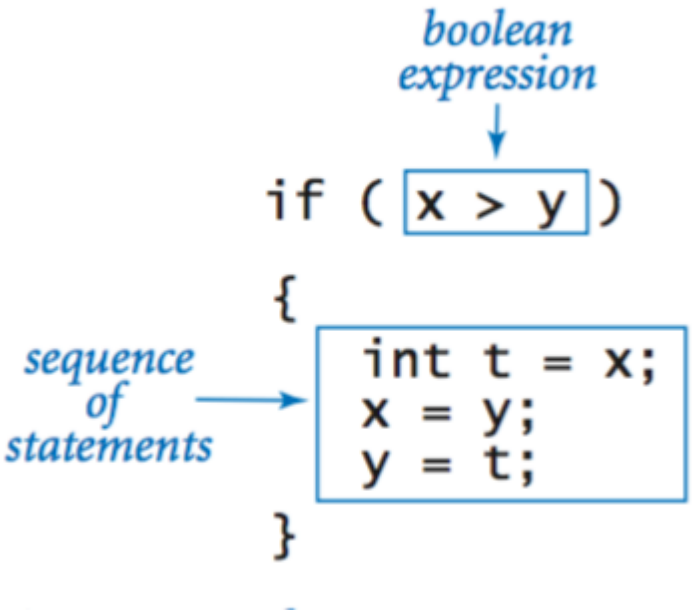
If and if-else statements.



absolute value	`if (x < 0) x = -x;`

put the smaller value in x and the larger value in y	`if (x > y)`
	`{`
	`  int t = x;`
	`  x = y;`
	`  y = t;`
	`}`

maximum of x and y	`if (x > y) max = x;` `else  max = y;`
error check for division operation	`if (den == 0) System.out.println("Division by zero");` `else  System.out.println("Quotient = " + num/den);`

error check for quadratic formula	`double discriminant = b*b - 4.0*c;`
	`if (discriminant < 0.0)`
	`{`
	`  System.out.println("No real roots");`
	`}`
	`else`
	`{`
	`  System.out.println((-b + Math.sqrt(discriminant))/2.0);`
	`  System.out.println((-b - Math.sqrt(discriminant))/2.0);`
	`}`

Nested if-else statement.

```
if      (income <      0) rate = 0.00;
else if (income <   8925) rate = 0.10;
else if (income <  36250) rate = 0.15;
else if (income <  87850) rate = 0.23;
else if (income < 183250) rate = 0.28;
else if (income < 398350) rate = 0.33;
else if (income < 400000) rate = 0.35;
else                      rate = 0.396;
```

Useful Loops

compute the largest power of 2 less than or equal to n	`int power = 1;`
	`while (power <= n/2)`
	`  power = 2*power;`
	`System.out.println(power);`

compute a finite sum $(1 + 2 + \ldots + n)$	`int sum = 0;`
	`for (int i = 1; i <= n; i++)`
	`  sum += i;`
	`System.out.println(sum);`

compute a finite product $(n! = 1 \times 2 \times \ldots \times n)$	`int product = 1;`
	`for (int i = 1; i <= n; i++)`
	`  product *= i;`
	`System.out.println(product);`

print a table of function values	`for (int i = 0; i <= n; i++)` `  System.out.println(i + " " + 2*Math.PI*i/n);`

compute the ruler function (see PROGRAM 1.2.1)	`String ruler = "1";`
	`for (int i = 2; i <= n; i++)`
	`  ruler = ruler + " " + i + " " + ruler;`
	`System.out.println(ruler);`

Useful Array Operations

create an array with random values

```
double[] a = new double[n];
for (int i = 0; i < n; i++)
   a[i] = Math.random();
```

print the array values, one per line

```
for (int i = 0; i < n; i++)
   System.out.println(a[i]);
```

find the maximum of the array values

```
double max = Double.NEGATIVE_INFINITY;
for (int i = 0; i < n; i++)
   if (a[i] > max) max = a[i];
```

compute the average of the array values

```
double sum = 0.0;
for (int i = 0; i < n; i++)
   sum += a[i];
double average = sum / n;
```

reverse the values within an array

```
for (int i = 0; i < n/2; i++)
{
   double temp = a[i];
   a[i] = a[n-1-i];
   a[n-i-1] = temp;
}
```

copy sequence of values to another array

```
double[] b = new double[n];
for (int i = 0; i < n; i++)
   b[i] = a[i];
```