

double b = 32 32 is a literal int stored as 32,0
 bc variable b is double

Literals are the fixed values used throughout the code. Literals have data types. For example:

double z = -1.8; -1.8 is a literal double. z is the variable
 boolean bob = false; false is the literal bob is the variable
 true is the other one

Do the first set of exercises.

Operations with Integers. The result of any operation involving two ints (whether variables or literals) is an int. When dividing two ints, the result is an int. only 4 bytes used so whole numbers!

int a; a = 29 / 10; What is the value of a? $\begin{array}{r} 2 \\ 10 \overline{) 29} \\ \underline{20} \\ 9 \end{array}$ ← ans 9 ← ignore	int b; b = 23 / 4; What is the value of b? $\begin{array}{r} 5 \\ 4 \overline{) 23} \\ \underline{20} \\ 3 \end{array}$ 3 ← ignore	double c; c = 387 / 100; What is the value of c? $\begin{array}{r} 3 \\ 100 \overline{) 387} \\ \underline{300} \\ 87 \end{array}$ → 4 bytes only so 3 87 ← ignore
--	---	--

Mixed Expressions. If an operation involves a double, then the result is a double.

int a = 87 / 10; int division	What is the value of a? 8
double b = 87 / 10.0; double division	What is the value of b? 8.7
double c = 87 / 10;	What is the value of c? 8.0 bc stored as double
int d = 8; double e = 10; int f = (5 + d) / e; double division	There is a problem and this code will not run. Why not. we have double division but cannot store 8 bytes into 4
double g = 10.0 * 76 / 100;	What is the value of g? 7.6
double h = 10.0 * (76 / 100);	What is the value of h? 0.0
double i = 3.21 + 24 / 9;	What is the value of i? 5.21

Displaying/Printing. When we want to display/print to the screen, there are two expressions we use:

System.out.println (an expression); After displaying, the cursor moves to the next line
 System.out.print (an expression); After displaying, the cursor stays on the current line

For example:

```
int n = 20;
n = n + 10 / 2;
System.out.print( "one fish " );
System.out.println( "n is " + n );
System.out.println( "two fish" );
```

Show what is displayed (pay attention to line breaks).

one fish n is 25

two fish

"one fish", "n is", and "two fish" are examples of string literals.

classes start w/ capitals

Do the second set of exercises.

* add in the coding activities in Edhesive

Scanner Objects and Simple Programs. To read in data from the keyboard, we will create an object of the Scanner class. For example:

import java.util.Scanner; *Common Error*
This import statement is required to use the Scanner class

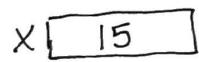
```

public class Example {
    public static void main(String[] args) {
        Scanner kb = new Scanner( System.in );
        System.out.println( "Enter an integer:" );
        int x = kb.nextInt();
        System.out.println( "You entered " + x );
        double y= kb.nextDouble();
        System.out.println( "You entered " + y );
    }
}

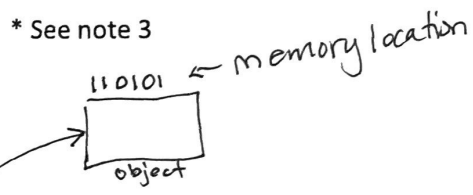
```

datatype (pointing to Scanner), *Var name* (pointing to kb), *parameters* (pointing to System.in)

objectname.method()



Common Error
* See note 1
* See note 2
you must have this so user knows to enter a value!
opens window



1. This statement declares a Scanner object. The expression on the right side of the assignment operator creates a Scanner object. The name of the variable can change but keep everything else the same.

2. The expression to the right of the assignment operator calls the nextInt method which retrieves the first integer entered by the user. This value is then assigned to the variable.

3. Use the next Double method when you expect the user to enter a double.

We will discuss objects and classes in more detail in the next unit. For now you only need to know how to create a Scanner object and use it to read ints and doubles.

Do the third set of exercises.

truncates the decimal portion.

Casting is the process of explicitly converting one data type to another. If you cast a double to an int, it is rounded toward zero. The casting operator has higher precedence than multiplication but lower than parentheses.

a 7

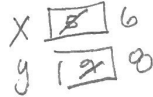
int a; a = (int) 7.8;	What is the value of a? 7
int b; b = (int) (-6 + 0.2);	What is the value of b? -5
int c; c = (int) 2.7 + 0.6;	This does not compile. Why? still a double
int d = 39; double e = (double) d / 10;	What is the value of e? 3.9

Increment and Decrement Operators.

You can increase the value of an int or double by using the increment operator (++). To decrease its value, use --. For example:

++x
increments immediately

```
int x = 5;
x++; x = x + 1;
int y = 9;
y--;
System.out.println(x + ", " + y);
```



add 1 when statement ends

```
int y;
int x = 5;
y = x++; y 5 x 6
6, 8
```

In this course we will only use the increment and decrement operators in stand-alone expressions. They will never be used as part of a larger expression in this course. For example:

```
int x = 1;
int y = 3 * x++;
System.out.println(x + ", " + y);
```

← the increment is post decrement bc it decrements after stmt
We do not do this in AP CS A
Surprisingly, this prints 2, 3
*2, 6 if we did int y = 3 * ++x;*

Using the increment and decrement operators in an expression (1) makes the expression harder to evaluate and (2) leads us into topics that are not part of our curriculum.

Compound Assignment Operators

Java sometimes uses "short cuts" for certain common statements

Basic Version	Alternative Version
int x = 7; x = x + 4;	int x = 7; x += 4;
int y = 14; y = y - 2;	int y = 14; y -= 2;

There is also *=, /=, and %=

Do the fourth set of exercises.

The Modulus Operator. The mod operator (a.k.a. the remainder operator) is the percent sign (%). It is used to find the remainder of a division operation. For example:

```
int x = 14 % 5;
int y = 24 % 6;
int z = 8 % 10;
```

4	
0	
8	10 8
	$\frac{0}{8} \leftarrow \%$

do long division

$$\begin{array}{r} 2 \\ 5 \overline{) 14} \\ \underline{10} \\ 4 \end{array} \leftarrow 14 / 5$$

$$\frac{0}{8} \leftarrow 14 \% 5$$

The modulus operator can be very useful. For example, suppose a store sells soft pretzels for 50 cents each and \$5 for a dozen. Here's a program that calculates the cost.

```
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        Scanner in = new Scanner( System.in );
        System.out.println( "How many pretzels do you want? " );
        int num = in.nextInt();

        double cost = 5 * ( num / 12 ) + 0.5 * ( num % 12 );
        System.out.println( "The cost is $" + cost );
    }
}
```

12 → \$5
1 → .50

Here is another sample program where the mod operator is useful. It converts minutes into hours and minutes.

```
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        Scanner s = new Scanner( System.in );
        System.out.println( "How many minutes? " );
        int time = s.nextInt();
        int hrs = time / 60;
        int m = time % 60;
        System.out.println( time + " min = " + hrs + " hour(s) and " + m + " minute(s) );
    }
}
```

If the user enters 73,
 what is hrs? 1
 what is m? 13

If the user enters 51,
 what is hrs? 0
 what is m? 51

Do the fifth set of exercises.