String literal “”

“123 Main Street”

“x”

Outputs a string

System.out object represents a **destination** (the monitor)to where output is sent

System.out.println (“Whatever you are, be a good one”);

Print does not advance to next line like println does

Print = one, two three

Println= one

Println= two

Println= three

Plus sign (+) can be used to concatenate two lines together

“peanut butter” + “jelly”

If both operands are stings or one is string and one is number, concatenation performs

If both operands are numeric, it adds them

+ operator is evaluated left to right

Parentheses can be used to force the order

**VARIABLES**

Name for location in memory

Used to store data

-like numbers and letters

Must be declared by specifying the name and type of information it will hold

Int total ;

int < data type

total; <variable name

int count, temp, result; < three variables declared in one line

camel naming

int radius; int speedRacing

**Variable Initialization**

Can be given an initial value

Can also be changed later with a new declaration

Int sum = 0;

Int base = 32, max= 149;

Whenever a variable is referenced the current value is used

**Assignment**

When the value of a variable is changed

Assignment operator is the equal sign =

**Total = 55;**

New value overwrites old value

**Constants**

Holds the same value throughout out the execution

Naming uppercase for constants

Use *final* modifier to declare a constant

final int MIN\_HEIGHT = 69;

**Primitive Data Types**

Eight primitive types

Four are integer

**byte,shot,int,long**

Two represent floating point

**Float, double**

One character type- stores a single character ‘x’ ‘d’ ‘t’

**Char**

One Boolean type- represent true or false condition

Can be used to represent any two states, such as lightbulb, on and off

**Boolean**

Int, double, char, boolean== most common data types

**Expression**

A combination of one or more operators

Similar to high school mathematics

* If both operands to the division operator (/) are integers, then the result is an integer
  + Fractional part is discarded
  + Integer rounding results in truncation, not rounding

Remainder operator (%) returns the remainder after dividing the second operand into the first

**Operator Precedence**

Multiplication, division and remainder come before addition subtraction and string concatenation

Executes left to right but parentheses take precedence