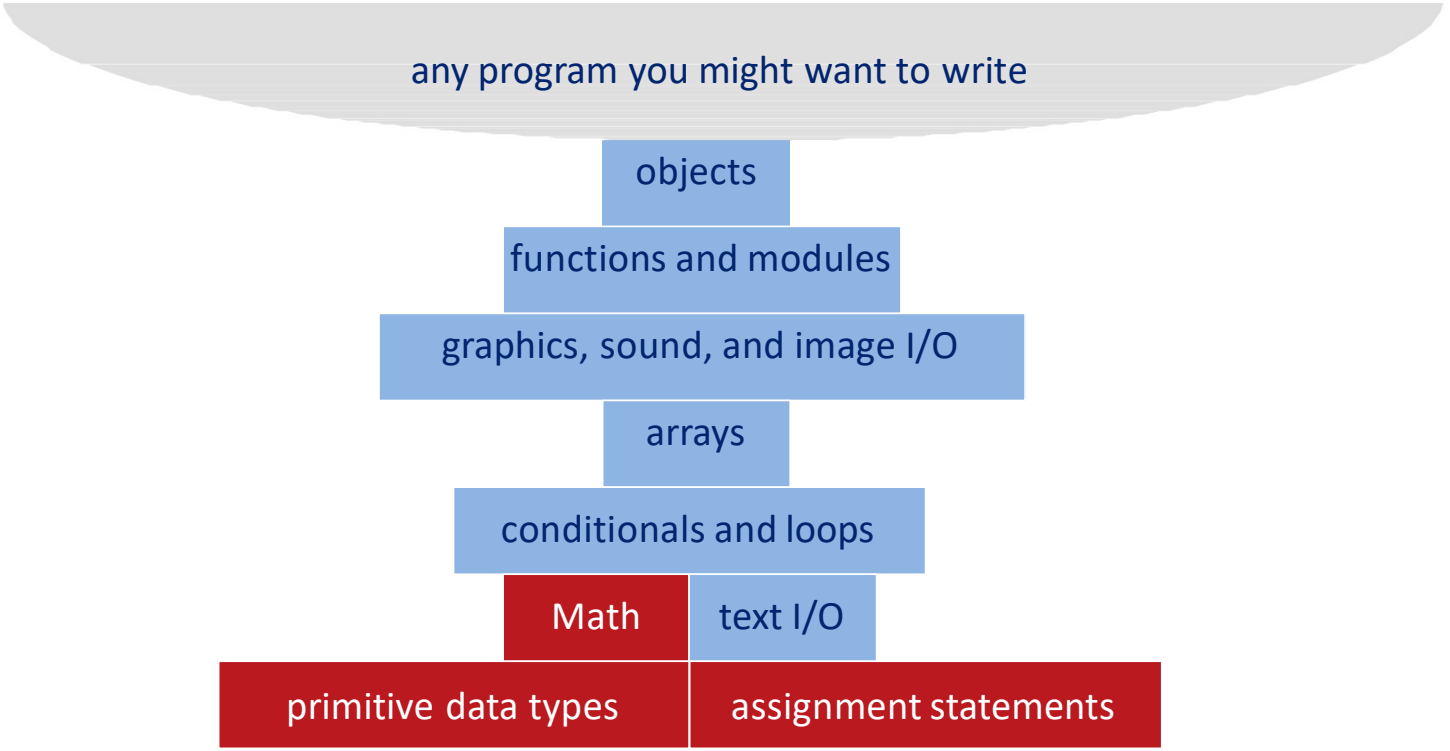
Variables

Penn Engineering

A Foundation for Programming



Penn Engineering

Variables

- A name to which data can be assigned
- A variable is declared as a specific data type
- You can assign a value to a variable when you declare it, but this is optional.
- Names must begin with a lowercase letter, '_' or '$' and can contain letters, digits, '_' and '$'

```
boolean bReady = true;
int i;
int j = 12;
color _red = color(255, 0, 0);
String name123 = "Fred";
```

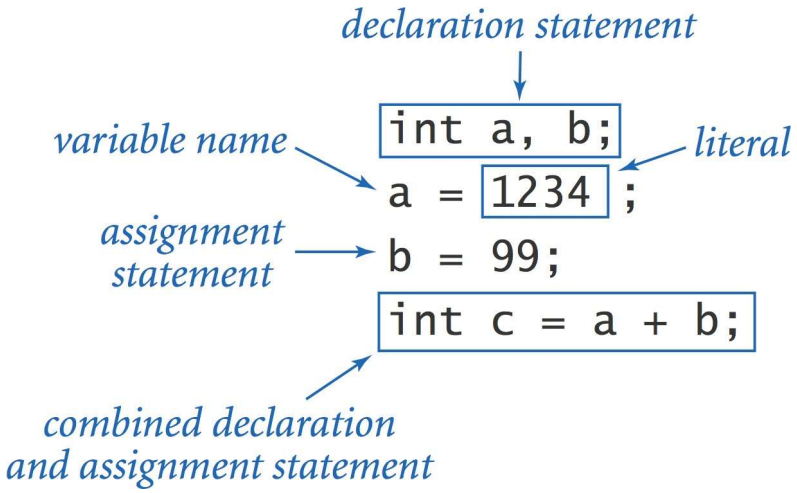
Components of a Variable Declaration	Example
Data Type	boolean
Name	bReady
Assignment (OPTIONAL)	= true;

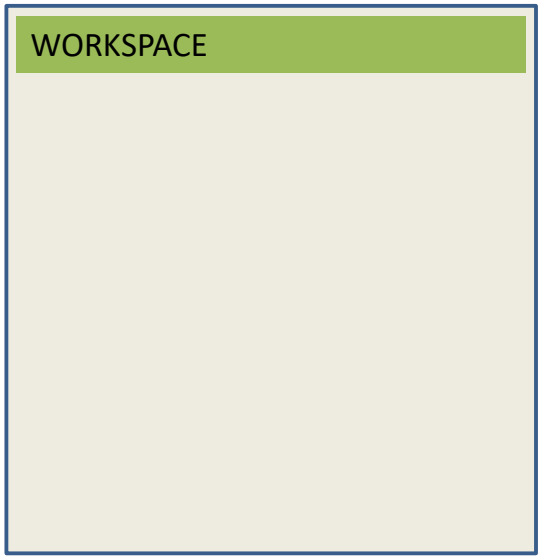
Penn Engineering

Variable Uses

- Use a value throughout your program,
  - but allow it to be changed
- As temporary storage for a intermediate computed result
- … etc

Variables and Types

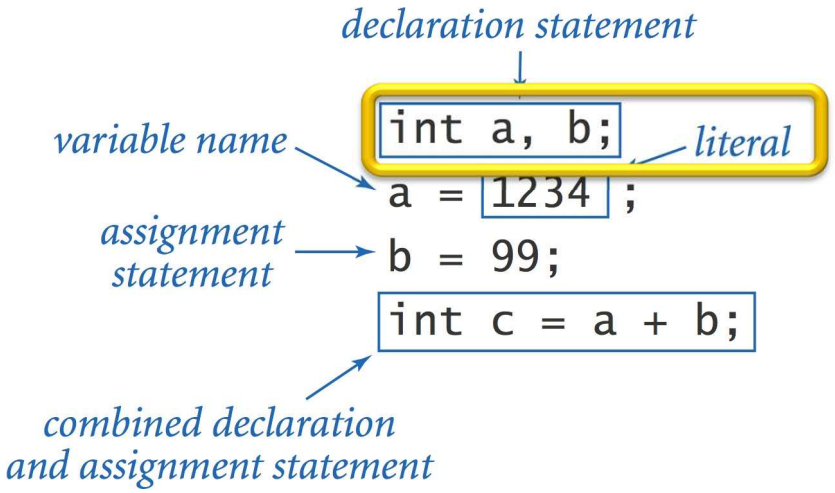


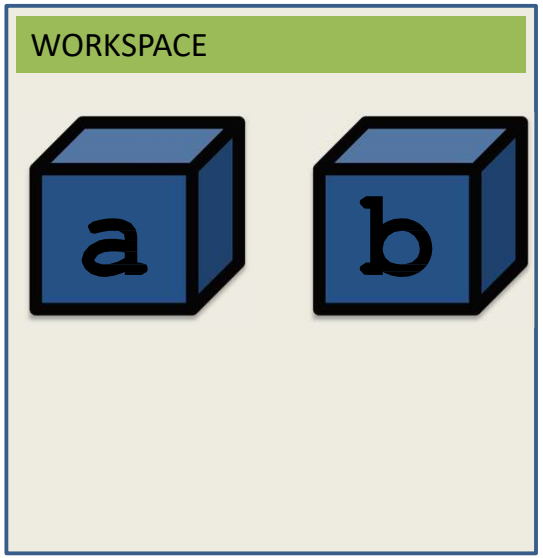


"int" means that the variable will <u>always</u> hold an integer

Penn Engineering

Variables and Types

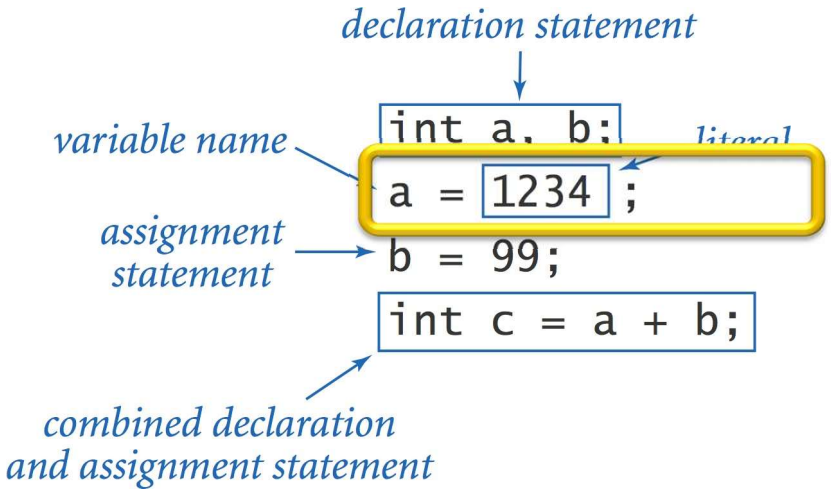


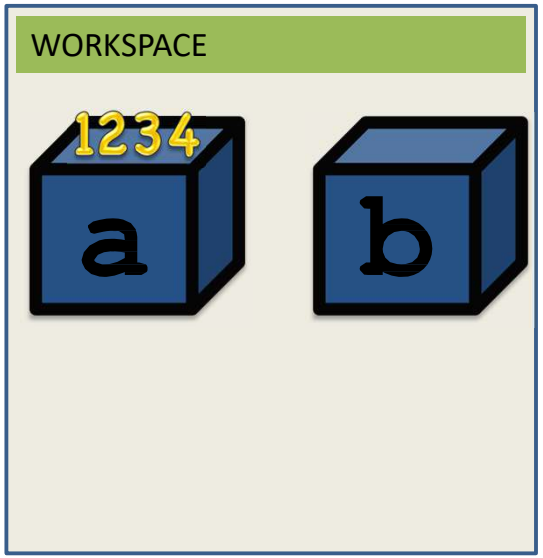


"int" means that the variable will <u>always</u> hold an integer

Penn Engineering

Variables and Types

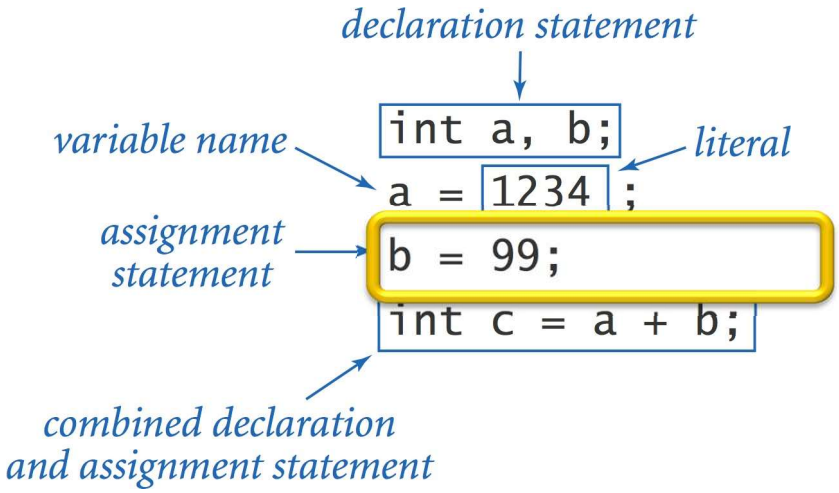


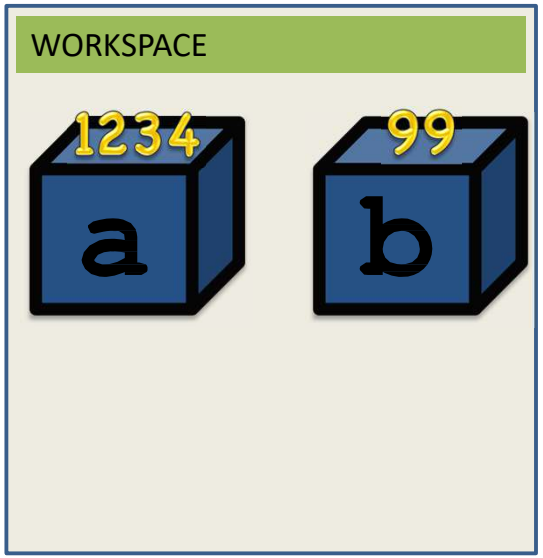


"int" means that the variable will always hold an integer

Penn Engineering

Variables and Types

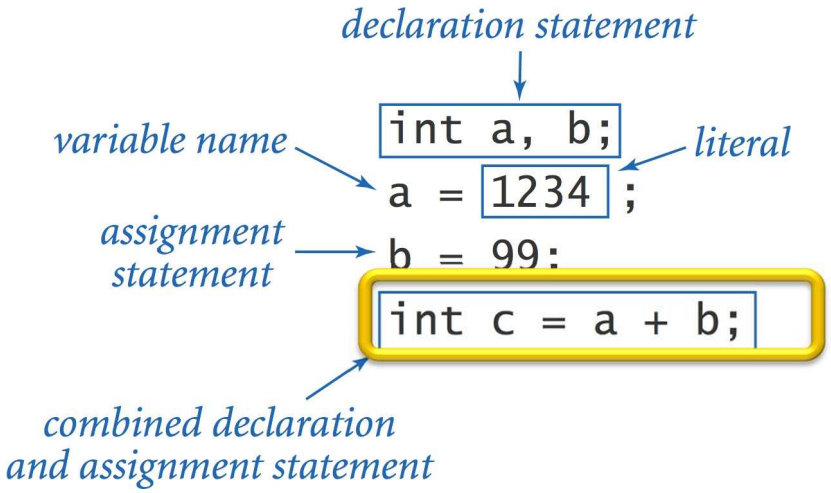


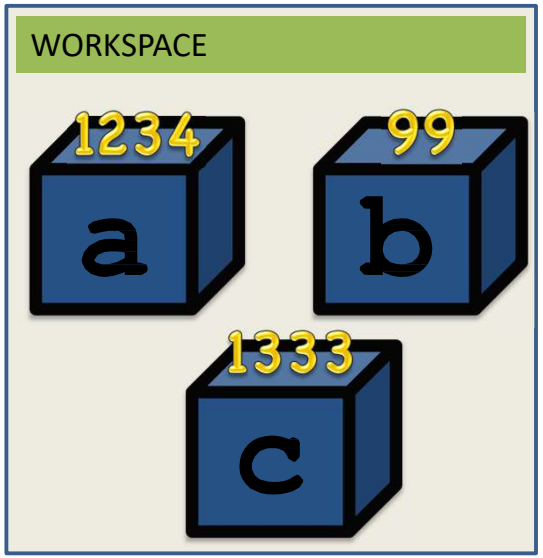


"int" means that the variable will always hold an integer

Penn Engineering

Variables and Types





"int" means that the variable will always hold an integer

Penn Engineering

Variable Scope

***Variable scope*:**

- That set of code statements in which the variable is known to the compiler
- Where it can be referenced in your program
- Limited to the ***code block*** in which it is defined
  - A ***code block*** is a set of code enclosed in braces (***{}***)

```
1 public class Main {
2     public static void main(String[] args) {
3         int shapeCounter = 0;
4         PennDraw.setCanvasSize(500, 500);
5         PennDraw.clear(PennDraw.BLUE);
6         PennDraw.setPenColor(0, 190, 0);
7         // draw a shape
8         PennDraw.filledRectangle(0.5, 0.25, 0.5, 0.25);
9         // add one to the count of shapes I've drawn
10        shapeCounter = shapeCounter + 1;
11    }
12 }
13
```

shapeCounter is a variable that is in scope only in the code block between **line 2** and **line 11**.

Penn Engineering

Primitive Data Types

Type	Range	Default	Bytes
boolean	{ true, false }	false	?
byte	{ 0..255 }	0	1
int	{ -2,147,483,648 ... 2,147,483,647 }	0	4
long	{ -9,223,372,036,854,775,808 ... 9,223,372,036,854,775,807 }	0	8
float	{ -3.40282347E+38 ... 3.40282347E+38 }	0.0	4
double	*much larger/smaller*	0.0	8
char	*a single character* 'a', 'b', …	'\u0000'	2

Penn Engineering

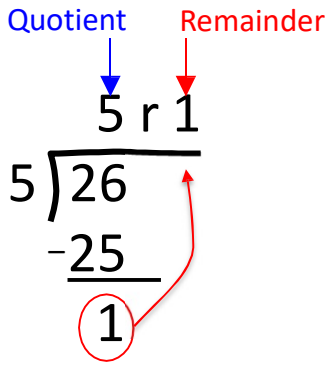
`int`: Integers (whole numbers)

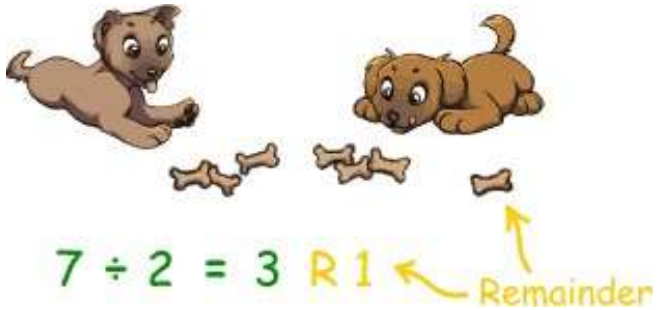
`+`, `-`, `*`, `/`, `%` (modulo), `()`, `Integer.parseInt()`

Expression	Result?
`5 + 3`	
`5 - 3`	
`5 * 3`	
`5 / 3`	
`5 % 3`	
`5 % -3`	
`1 / 0`	
`3 * 5 - 2`	
`3 + 5 / 2`	
`3 - 5 / 2`	
`(3 - 5) / 2`	
`3 - (5 - 2) / 2`	
`Integer.parseInt("3")`	
`Integer.parseInt(3)`	

Penn Engineering

Modulo Operator (%)





Division gives the quotient:

26 / 5 == 5

Modulo gives the remainder:

26 % 5 == 1

Example: Determining whether an integer `n` is even or odd:

```
boolean isEven = (n % 2 == 0);
```

Penn Engineering

Dog image from http://www.mathsisfun.com/

`double`: Floating-Point (fractions)

`+`, `-`, `*`, `/`, `%` (modulo), `()`, `Double.parseDouble()`

Expression	Result?
3.141 + 0.03	
6.02e23 / 2.0	
5.0 / 3	
(int) 5.0 / 3	
5.0 / (int) 3	
10.0 % 3.141	
1.0 / 0.0	
-1.0 / 0.0	
0.0 / 0.0	
Math.sqrt(2)	
Math.sqrt(-1)	
Math.sqrt(2) * Math.sqrt(2)	
Math.PI	
Math.pi	

Penn Engineering

Java Math Library (Excerpts)

`public class Math`

`double abs(double a)`	*absolute value of a*
`double max(double a, double b)`	*maximum of a and b*
`double min(double a, double b)`	*minimum of a and b*

Note 1: `abs()`, `max()`, *and* `min()` *are defined also for* `int`, `long`, *and* `float`.

`double sin(double theta)`	*sine function*
`double cos(double theta)`	*cosine function*
`double tan(double theta)`	*tangent function*

Note 2: Angles are expressed in radians. Use `toDegrees()` *and* `toRadians()` *to convert.*
Note 3: Use `asin()`, `acos()`, *and* `atan()` *for inverse functions.*

`double exp(double a)`	*exponential* (e^a)
`double log(double a)`	*natural log* ($\log_e a$, *or* $\ln a$)
`double pow(double a, double b)`	*raise a to the bth power* (a^b)
`long round(double a)`	*round to the nearest integer*
`double random()`	*random number in* [0, 1)
`double sqrt(double a)`	*square root of a*
`double E`	*value of e (constant)*
`double PI`	*value of* π *(constant)*

Penn Engineering

`char`: Single Characters

Single characters are stored as (small) integers!

Expression	Result?
'A'	
'A' + 0	
(int) 'A'	
(char) 65	
(int) 'a'	
(int) '0'	
'3' – '0'	

Penn Engineering

`boolean`: True/False

`true`, `false`, `==`, `!=`, `<`, `>`, `<=`, `>=`, `&&` (and), `||` (or), `!` (not)

Expression	Result?
true	
!false	
'A' == 'a'	
Math.PI != 3.14	
'a' > 'b	
1.7 <= (17 / 10)	
true && true	
true && false	
false && false	
true || true	
true || false	
false || false	
(1 < 3) && (3 == (6 / 2))	
(1 >= 3) || !(3 == (6 / 2))	

Penn Engineering

Data Type Conversion

- Some variable types can be converted to other types via **casting**

```
double f = 10.0;
int i = (int) f;
System.out.println(f);
System.out.println(i);

//i = f;          // Throws a runtime error
```

Penn Engineering

More Complex Data Types

Type	Range	Default	Bytes
String	a series of chars in quotes “abc”	null	?
PImage	an image	null	?
PFont	a font for rendering text	null	?
…			

Penn Engineering