

# 15-121 – Spring, 2018

## Intro to Data Structures

Lecture #2 – Java Basics (Types, Expressions)  
January 17, 2018

Mark Stehlik

# Outline for Today

---

- Office hours
  - Darien: Tue 4:30-6:30, Sun 2:00-6:00; Gates 3
  - Me: Mon 1:30-4:30, Fri 11:00-1:00; Gates 6205
- Java Types (operators/precedence)
  - Types (primitive/reference)
  - Variables: begin with letter (cap for class, lower for variable/method; then letters, numbers, and \_ )
  - Expressions

# Creating and Executing programs

---

- Source code → byte code → execution/run  
                    compilation          jvm
- Python is *interpreted* (translated as you go); Java is *compiled* (translated in its entirety)
- An example to illustrate the difference:

```
def test():  
    x = 12  
    if 9 < 10:  
        print(x)  
    else:  
        print(y)  
test()
```

# Creating and Executing programs

---

- Comments
  - Purpose
  - Single line vs. multi-line `//` vs. `/* ... */`
- Errors
  - Syntax/Compile-Time
  - Runtime – always result in crash
  - Semantic/Logic
- Java is statically-typed (unlike Python)

# Primitive Types

- Primitives (8, but we'll use 4 primarily)
  - int (4 bytes == ? bits; max value == Integer.MAX\_VALUE)
  - double (8 bytes; max value == Double.MAX\_VALUE)
  - boolean (2 values: true and false)
  - char (2 bytes; max value?)
  - The others: byte (8 bits), short (16), long (64), float (32)
- Bits to values:
  - n bits can store  $2^n$  values (e.g., 8 bits can store 256 values)
  - Unsigned values range from 0 to  $2^n - 1$
  - Signed integers range from  $-(2^{n-1})$  through 0 to  $2^{n-1} - 1$
  - So 32 bit ints range from -2,147,483,648 to 2,147,483,647

# Basic Types (operators)

- Primitive operators
  - int: `+`, `-`, `*`, `/` (div), `%` (mod), increment/decrement (`++`, `--`), assignment (`=`, but also `+=`, `-=`, etc.)  
`Math.abs()`, `Math.min()`, `Math.max()`
  - double: `/` (real division), never compare doubles for equality!  
`Math.pow()`, `Math.sqrt()`
  - boolean: equality (`==`, `!=`), relational (`<`, `>`, `<=`, `>=`), logical (`!`, `&&`, `||`)  $\rightarrow$  (not, and, or)
  - char: single quote-delimited, operators? (not so much)
  - String: double quote-delimited
- Casting

# Basic Types (operator precedence)

---

- Precedence
  - postfix (expression++ expression--)
  - unary/prefix (++expression --expression -expression !)
  - \* / %
  - + -
  - < > <= >=
  - == !=
  - &&
  - ||
  - ? :
  - assignment

# Basic Types (sort of – String)

---

- String (not a primitive type, but a class)
  - String variables store references (addresses), not Strings!
  - Strings are immutable!
  - useful operations (methods):
    - charAt, compareTo, contains, equals (not ==), length, substring, trim
  - there are many more – check out the String API (the Java API is going to be your friend during the course!)



# ASCII table

Dec	Hex	Name	Char	Ctrl-char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
0	0	Null	NUL	CTRL-@	32	20	Space	64	40	@	96	60	`
1	1	Start of heading	SOH	CTRL-A	33	21	!	65	41	A	97	61	a
2	2	Start of text	STX	CTRL-B	34	22	"	66	42	B	98	62	b
3	3	End of text	ETX	CTRL-C	35	23	#	67	43	C	99	63	c
4	4	End of xmit	EOT	CTRL-D	36	24	\$	68	44	D	100	64	d
5	5	Enquiry	ENQ	CTRL-E	37	25	%	69	45	E	101	65	e
6	6	Acknowledge	ACK	CTRL-F	38	26	&	70	46	F	102	66	f
7	7	Bell	BEL	CTRL-G	39	27	'	71	47	G	103	67	g
8	8	Backspace	BS	CTRL-H	40	28	(	72	48	H	104	68	h
9	9	Horizontal tab	HT	CTRL-I	41	29	)	73	49	I	105	69	i
10	0A	Line feed	LF	CTRL-J	42	2A	*	74	4A	J	106	6A	j
11	0B	Vertical tab	VT	CTRL-K	43	2B	+	75	4B	K	107	6B	k
12	0C	Form feed	FF	CTRL-L	44	2C	,	76	4C	L	108	6C	l
13	0D	Carriage feed	CR	CTRL-M	45	2D	-	77	4D	M	109	6D	m
14	0E	Shift out	SO	CTRL-N	46	2E	.	78	4E	N	110	6E	n
15	0F	Shift in	SI	CTRL-O	47	2F	/	79	4F	O	111	6F	o
16	10	Data line escape	DLE	CTRL-P	48	30	0	80	50	P	112	70	p
17	11	Device control 1	DC1	CTRL-Q	49	31	1	81	51	Q	113	71	q
18	12	Device control 2	DC2	CTRL-R	50	32	2	82	52	R	114	72	r
19	13	Device control 3	DC3	CTRL-S	51	33	3	83	53	S	115	73	s
20	14	Device control 4	DC4	CTRL-T	52	34	4	84	54	T	116	74	t
21	15	Neg acknowledge	NAK	CTRL-U	53	35	5	85	55	U	117	75	u
22	16	Synchronous idle	SYN	CTRL-V	54	36	6	86	56	V	118	76	v
23	17	End of xmit block	ETB	CTRL-W	55	37	7	87	57	W	119	77	w
24	18	Cancel	CAN	CTRL-X	56	38	8	88	58	X	120	78	x
25	19	End of medium	EM	CTRL-Y	57	39	9	89	59	Y	121	79	y
26	1A	Substitute	SUB	CTRL-Z	58	3A	:	90	5A	Z	122	7A	z
27	1B	Escape	ESC	CTRL-[	59	3B	;	91	5B	[	123	7B	{
28	1C	File separator	FS	CTRL-\	60	3C	<	92	5C	\	124	7C	
29	1D	Group separator	GS	CTRL-]	61	3D	=	93	5D	]	125	7D	}
30	1E	Record separator	RS	CTRL-^	62	3E	>	94	5E	^	126	7E	~
31	1F	Unit separator	US	CTRL-`	63	3F	?	95	5F	_	127	7F	DEL