CS 302: Introduction to Programming

Lecture 4

THE UNIVERSITY of WISCONSIN MADISON

Example 1: Integer Arithmetic

Hey, explain the code this time!!! Open Eclipse ASAP~~~

THE UNIVERSITY of WISCONSIN MADISON

Example 2: Integer Division

Alt + Tab to switch to Eclipse

THE UNIVERSITY of WISCONSIN MADISON

Example 3: FloatingPointArithmetic

- Get started

THE UNIVERSITY of WISCONSIN MADISON

Example 4: Powers && Roots

- Get started

THE UNIVERSITY of WISCONSIN MADISON

Example 5: Input Example

- Martian weight converter
  - Input: Name, Weight
  - Output: Weight on Mars

THE UNIVERSITY of WISCONSIN MADISON

Practice a program combining arithmetic and Scanner class

- Now it's your turn
- Make a Java program that reads a number of cents and then prints out correct change, in US coinage

THE UNIVERSITY of WISCONSIN MADISON

A Note on Arithmetic Conventions

- Often many ways to write the same thing
  - x = x + 1; vs x++; vs x += 1; vs ++x;
  - x = x – 1; vs x--; vs x -= 1; vs --x;
  - x += 5; vs x = x + 5;
  - x /= 3; vs x = x / 3;

```
int x = 5, y = 5;
System.out.println(++x); // outputs 6
System.out.println(x); // outputs 6
System.out.println(y++); // outputs 5
System.out.println(y); // outputs 6
```

Strings

- Sequence of characters
- Reference type (non-primitive)
- Specified by double quotes (")
- Can have length 0 – empty string = ""
- Examples:
  - String name = "Dan";
  - String className = "CS302: Intro to Programming";

THE UNIVERSITY of WISCONSIN MADISON

›

String Operations

- Concatenation (+)
  - Have already seen in our output statements
  - Ex: String name = "Ned" + " Stark";
  - String className = "cs";
  - int classNum = 302;
  - className = className + classNum; //className is now: “cs302”
- Length
  - String name = "Luke Skywalker";
  - int length = name.length(); //length = 14
    - Remember identifier.methodName()

THE UNIVERSITY of WISCONSIN MADISON

Converting Strings to Numbers

- String ⟶ int
  - Integer.parseInt([String])
  - String aNumber = "5";
  - int x = Integer.parseInt(aNumber);
- String ⟶ double
  - Double.parseDouble([String])
  - double y = Double.parseDouble("2.2");

THE UNIVERSITY of WISCONSIN MADISON

Chars

- Single character
- Specified by single quotes (')
- Has numeric value
- Ex.

```
char myChar = 'a';
System.out.println(myChar); //will print out: a
myChar++;
System.out.println(myChar); //will print out: b
```

THE UNIVERSITY of WISCONSIN MADISON

ASCII Table Values

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040		Space	64	40	100	@	@	96	60	140	`	`
1	1	001	SOH (start of heading)	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	STX (start of text)	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	ETX (end of text)	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	EOT (end of transmission)	36	24	044	$	$	68	44	104	D	D	100	64	144	d	d
5	5	005	ENQ (enquiry)	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	ACK (acknowledge)	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	BEL (bell)	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	BS (backspace)	40	28	050	(	(	72	48	110	H	H	104	68	150	h	h
9	9	011	TAB (horizontal tab)	41	29	051	)	)	73	49	111	I	I	105	69	151	i	i
10	A	012	LF (NL line feed, new line)	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	VT (vertical tab)	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	FF (NP form feed, new page)	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	CR (carriage return)	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	SO (shift out)	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	SI (shift in)	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	DLE (data link escape)	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	DC1 (device control 1)	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	DC2 (device control 2)	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	DC3 (device control 3)	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	DC4 (device control 4)	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	NAK (negative acknowledge)	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	SYN (synchronous idle)	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	ETB (end of trans. block)	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	CAN (cancel)	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	EM (end of medium)	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	SUB (substitute)	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	ESC (escape)	59	3B	073	;	;	91	5B	133	[	[	123	7B	173	{	{
28	1C	034	FS (file separator)	60	3C	074	<	<	92	5C	134	\	\	124	7C	174			|
29	1D	035	GS (group separator)	61	3D	075	=	=	93	5D	135	]	]	125	7D	175	}	}
30	1E	036	RS (record separator)	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	US (unit separator)	63	3F	077	?	?	95	5F	137	_	_	127	7F	177	&#127;	DEL

```
int x = (int) 'a';
System.out.println(x); //output: 97
char myChar = (char) (x++);
System.out.println(myChar); //output: b
```

THE UNIVERSITY of WISCONSIN MADISON

charAt

- Method to find a specific character within a String
- Strings are 0-indexed
- Ex.
- String name = “Bill Clinton";
- char first = name.charAt(0); //first = ‘B'
- int length = name.length(); //length = ?
- char last = name.charAt(length – 1); //last = ‘n'
- What if I had done:

```
char last = name.charAt(length);
```

THE UNIVERSITY of WISCONSIN MADISON

Substrings

- What if I want to get part of a String?
- stringName.substring([start], [end])
  - Will include charAt(start)
  - Will include charAt(end - 1);
  - Will NOT include charAt(end)
  - Start, end, must be ints
- Remember the 0-indexed nature of Strings
- Ex.
- String name = “Barack Obama";
- String first = name.substring(0, 3);
- String last = name.substring(4);

THE UNIVERSITY of WISCONSIN MADISON

Example Time

- Show how to actually use all the methods we learned

THE UNIVERSITY of WISCONSIN MADISON

Practice with String

- Reads in a social security number (SSN), formatted as **XXX-XX-XXXX, where the X's** represent digits. Adds 1 to that number and prints the result in the same format.

THE UNIVERSITY of WISCONSIN MADISON

If statement

- What if I want to make a decision?
- Parts:
  - Boolean expression (a statement that is either true or false)
  - Code
- Ex.

```
if (5 > 1)
{
    System.out.println("Five is greater than 1");
}
```

THE UNIVERSITY of WISCONSIN MADISON

Comparing Numbers: Relational Operations

- ==
  - Is something equal to something else
  - if (a == b)
- >
  - Greater than
- <
  - Less than
- >=
  - Greater than or equal to
- <=
  - Less than or equal to
- !=
  - Not equal
- Precedence
  - Lower precedence than arithmetic operators
  - Ex. what does (3 + 2 < 5) evaluate to?

Comparing Strings

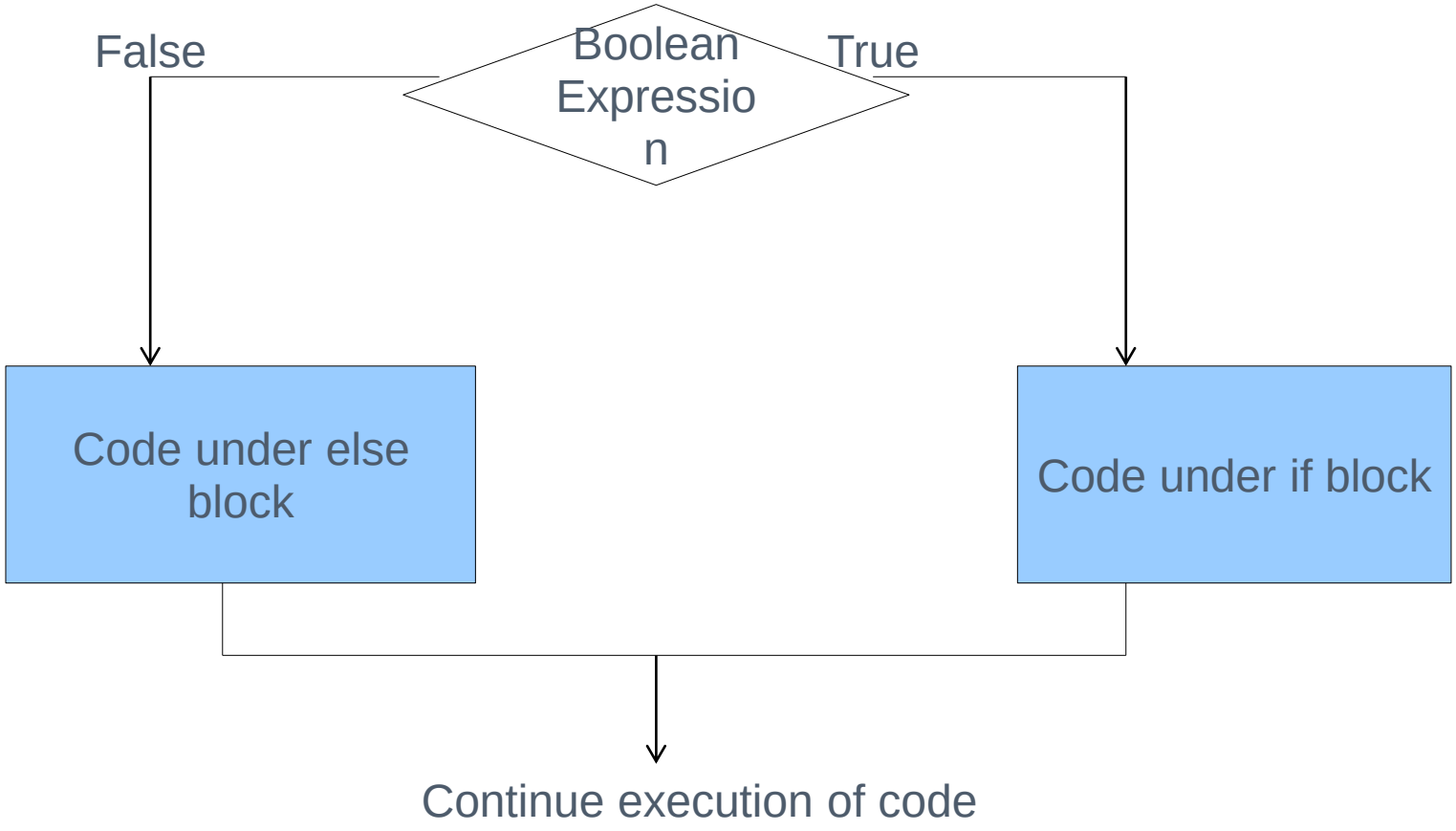
- Do NOT use ==
- Strings are reference variables, not primitives
- Instead use .equals() and .equalsIgnoreCase()
- Also .compareTo()
  - Returns an int
- Format:
  - stringOne.equals(stringTw

THE UNIVERSITY of WISCONSIN MADISON

```
String foo = "abcdef";
String bar = "ABCDEF";
if (foo.equals(bar))
{
    System.out.println("foo equals
    bar");
}
if (foo.equalsIgnoreCase(bar))
{
    System.out.println("foo equals
    bar if you ignore the case");
}
```

Else

- Else
  - Code that executes if the boolean expression was false



THE UNIVERSITY of WISCONSIN MADISON

Else Example

```
String foo = "abcdef";
String bar = "ABCDEF";
if (foo.equals(bar))
{
    System.out.println("foo equals bar");
}
else
{
    System.out.println("foo doesn't equal bar");
}
```

THE UNIVERSITY of WISCONSIN MADISON