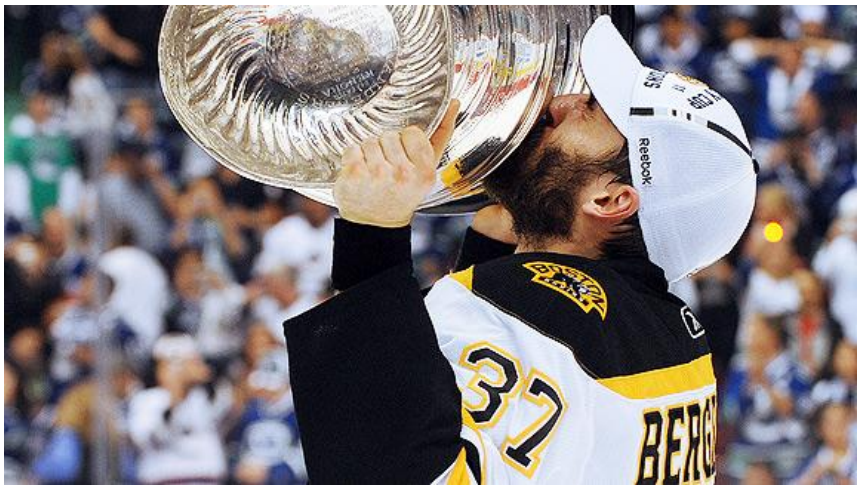
CS 302: INTRODUCTION TO PROGRAMMING

Lectures 5&6



THE UNIVERSITY of WISCONSIN MADISON

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()

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
```

CHARAT

- Method to find a specific character within a String
- Strings are 0-indexed
- Ex.
- String name = "Dan Szafir";
- char first = name.charAt(0); //first = 'D'
- int length = name.length(); //length = ?
- char last = name.charAt(length – 1); //last = 'r'
- What if I had done:

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

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 = "Dan Szafir";
- String first = name.substring(0, 3);
- String last = name.substring(4);

INDEXOF

- Opposite of charAt
- Finds the first occurrence of a char in a String

```
System.out.println("Enter your favorite team");
String name = in.nextLine(); //Assume Boston Bruins was
    entered
int spaceIndex = name.indexOf(' '); //spaceIndex = 6
String city = name.substring(0, spaceIndex);
```

- Will return -1 if the specified character was NOT in the String

STRING METHODS SUMMARY

- .length() - counts the number of chars in a String
- .charAt([index]) – returns the char at [index]
- .substring([start], [end]) – returns a String whose content is the character at [start] up to but not including the char at [end]
- .substring([start]) – returns a String whose content is the character at [start] through the end of the original String
- .indexOf([char]) – returns the first occurrence of [char] in the String, or -1 if it wasn't found

EXAMPLE CODE FOR USING METHODS FROM STRING CLASS

Switch to Eclipse

THE UNIVERSITY of WISCONSIN MADISON

(CHAP. 3)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?

THE UNIVERSITY of WISCONSIN MADISON

COMPARING STRINGS

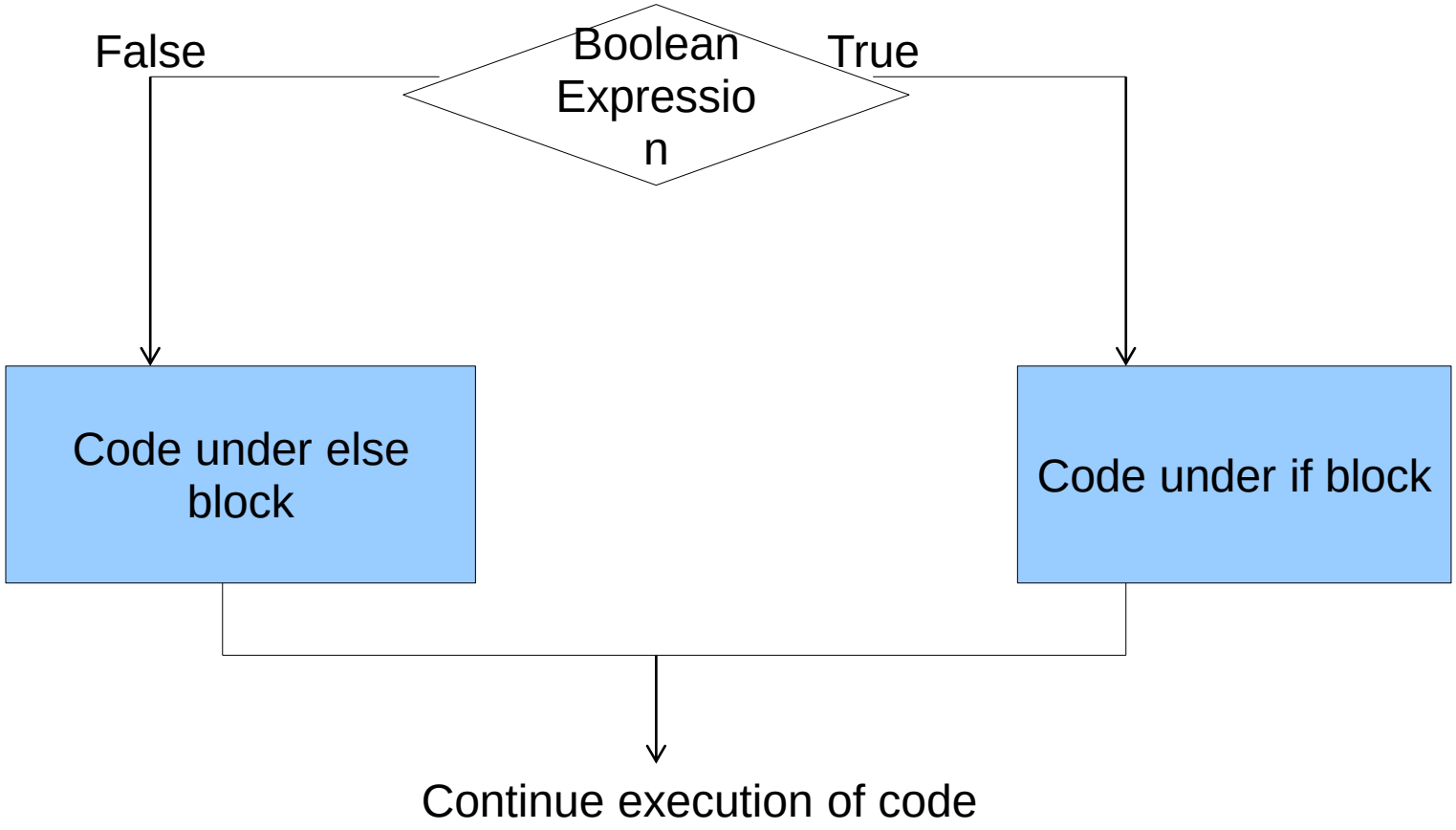
- Do NOT use ==
- Strings are reference variables, not primitives
- Instead use .equals() and .equalsIgnoreCase()
- Also .compareTo()
  - Returns an int
- Format:
  - stringOne.equals(stringTwo)

```
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");
}
```

THE UNIVERSITY of WISCONSIN MADISON

ELSE

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



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

ELSE IF

- What if I wanted to check more than one thing?
- if (test something)
- {
  //do something
- }
- else if (test something else)
- {
  //do something else
- }
- else
- {
  //what to do if both of those tests were false
- }

IF, ELSE IF, AND ELSE

- If
  - One or none
- If...Else
  - One or another
- If...Else If...Else
  - One of many
- The only thing you need is an if
  - Can have if and else ifs with no else
  - Can have if and else with no else ifs
  - Can have if alone
  - CANNOT have and else if or an else without a starting if

THE UNIVERSITY of WISCONSIN MADISON

SCOPE

- Determines in what context values and expressions are associated
- General Rule of Thumb:
  - Variables defined within a set of braces are only good within that set (and any nested sets)

PRACTICE 1

- Write a simple Log-In program:
  - Input: Username
  - Output:
    - If username matches a known username output: "Hello [username], good to see you again!"
    - Else: "Invalid Login Attempt"

PRACTICE 2

- Write a program according to these specifications:
  - Input: Day of the week, Year
  - Output:
    - If Sunday: “Yikes, tomorrow I have to work again :("
    - If Saturday: “Hooray, I can hang out with friends today :)"
    - If Monday: “Alas, I’m lecturing right now~~~ but cheer up"
    - Otherwise: "Just another weekday, let’s enjoy working"
    - If Year is evenly divisible by 4: "Leap year, we can all live 1 more day this year, isn’t that great?"

PRACTICE 3

- Write a program to convert numerical grades to letter grades:
  - Input: a numerical grade 0 – 100
  - Output:
    - Grade: 90 - 100: "A, you must have a great IQ score"
    - Grade: 80 – 90: "B, it's okay, but just don't tell your mom"
    - Grade 70 – 80: "C, got addicted to Diablo?"
    - Grade 60 – 70: "D, oops, the student passes out"
    - Grade 0 – 60: "F, no matter what, your instructor is just ruthless"
    - Otherwise: "Error, invalid grade"

THE UNIVERSITY of WISCONSIN MADISON

PRACTICE 4: MULTI-PLANET WEIGHT CONVERTER

THE UNIVERSITY of WISCONSIN MADISON

SWITCH STATEMENT

- Alternative to having if...else if... else if... else...
- Use if you are testing the same variable in each boolean expressior
- Ex.

```
int day = in.nextInt(); String dayName = "";
switch(day)
{
   case 1: dayName = "Sunday"; break; //same as if (day == 1)
   case 2: dayName = "Monday"; break; //else if (day == 2)
   case 3: dayName = "Tuesday"; break;
   //...more cases...
   default: dayName = ""; break; //default is a catchall like else
    statements
}
```

SWITCH STATEMENT

- Will "fall through" to the next case if there is no break:

```
switch(day)
{
   case 1: dayName = "Sunday";
   case 2: dayName = "Monday"; break;
   case 3: dayName = "Tuesday"; break;
   //...more cases...
   default: dayName = ""; break;
}
```

What is the value of dayName if day = 1?

SWITCH STATEMENT

- Sometimes you want to fall through

```
switch(day)
{
 case 1:
 case 7: dayType = "weekend"; break;
 case 2:
 case 3:
 case 4:
 case 5:
 case 6: dayType = "weekday"; break;
 default: dayType = "unknown"; break;
}
```

PRACTICE 5

- Use switch statement
- Take an int input for the nth day in a week (starting from Sunday)

- If (Sat or Sun), print out “It's a weekend”
- If (MonTuTh), print out “It's a weekday”
- If (Wed), print out “Halfway there”
- If (Fri), print out “It's almost the weekend”
- Otherwise, print out “Not a valid day (day's should be 1-7)”

THE UNIVERSITY of WISCONSIN MADISON

NESTED BRANCHES

- If statement within another if, else if, or else statement

```
if (I'm hungry)
{
    if (I feel like Italian)
    {
        Make spaghetti
    }
    else
    {
        Make a hamburger
    }
}
```

```
int temperature = in.nextInt();
int raining = in.nextInt(); //1 means yes, otherwise no
if (temperature > 70) {
   if (raining == 1) {
      S.o.pln("Wear shorts and bring an umbrella");
   }
   else {
      S.o.pln("Wear shorts and sunglasses");
  }
}
else {
   S.o.pln("It is indeed a typical WI weather");
}
```

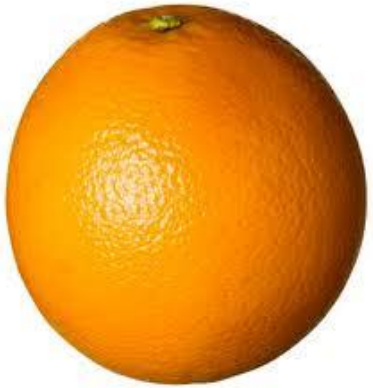
BOOLEAN VARIABLES

- Booleans are either true or false;
- Ex.
- boolean failed = false;
- if (failed) //same as if (failed == true)
- {

  //stop the program
- }

THE UNIVERSITY of WISCONSIN MADISON

INTRO TO BOOLEAN OPERATORS

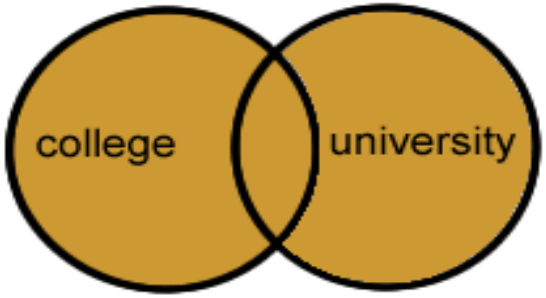


- I want an apple and an orange
- I want an apple or an orange

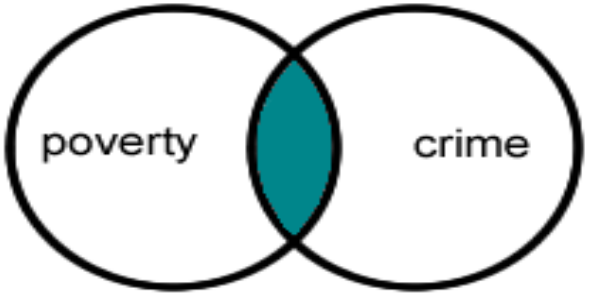
BOOLEAN OPERATORS

- Ways to combine boolean variables
- && = and
- || = or
- ! = not

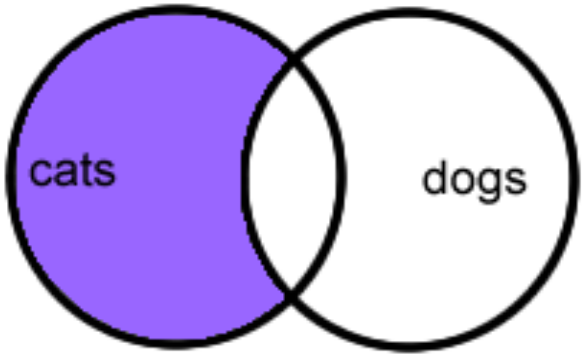
VISUALIZING BOOLEAN OPERATORS



college || university



poverty && crime



cats && !dogs

VERBALLY VISUALLIZING BOOLEAN OPERATORS

- "I carry an umbrella if it rains or snows"
  - if (rain || snow) : carry umbrella
- "I only wear shorts if its hot and sunny"
  - if (hot && sunny) : wear shorts

BOOLEAN OPERATORS AND NESTING

```
if (I'm hungry)
{
    if (I feel like Italian)
    {
        Make spaghetti
    }
    else
    {
        Make a hamburger
    }
}
```

```
if (I'm hungry && I feel like
    Italian)
{
        Make spaghetti
}
else if (I'm hungry)
{
        Make a hamburger
}
```

USE ! FOR NOT!

- Wrong

```
if (solved)
{ }
else
{
  //lots of code
}
```

- Right

```
if (!solved)
{
  //lots of code
}
//if(!solved) is the
   same
//as if (solved == false)
```

INPUT VALIDATION

- Use if statements to make sure the user input a valid value
- in.hasNextInt() - check if the user input an int
- in.hasNextDouble() – check if user input a double

PRACTICE

- More complicated Login Program
  - Input: Username, Password
  - If username matches a known username and password matches the corresponding password, output: *Giddy Up!*
  - Otherwise output: *Invalid Login*
  - Known Usernames / Passwords:
    - Jerry / porsche
    - George / Bosco

THE UNIVERSITY of WISCONSIN MADISON