The C Programming Language Chapter 3

(material from Dr. Michael Lewis, William & Mary Computer Science)

Overview

- Booleans
- Comparison Operators
- Conditional Expressions
- if Statement
- while/for/do Statements
- switch Statement

2

Python vs. C vs. C++ vs. Java For the most part the flow control in C is similar to that in Python, though C has a few constructs that Python lacks. Python true, false same as C same as C booleans True, raise true, failse or non-zero, zero same as C logical AND: and Kit same as C logical GR: or || same as C logical NOT not same as C same as C comparison operators: <, <=, ==, 1=, >=, > same same¹ same comparison operators Sentity 15 membenship in conditional expression expr2 if expr1 else expr1 if expr1
 membership
 is

 conditional operation (groups)
 sep:1 / equit (since);
 sep:1 / equit (since);
 same as C

 conditional
 if equit :
 if (equit)
 same as C

 conditional
 if equit :
 if (equit)
 same as C

 citit :
 eller :
 if (equit)
 same as C

 citit :
 eller :
 eller :
 same as C

 setter :
 eller :
 eller :
 eller :

 ford :
 tradit :
 eller :
 same as C
 same as C same as C same as C aame as C same as C range for for (i:object) foreach: for (i:object) for i in iterable: for (expr1; expr2; expr3) {} same as C same as C same as C + break and branch same as C + continue and branch same as C r (expr1; expr2; expr3) () same as C do (} while (expr); same as C break; same as C continue; same as C 10 break same as C choice switch same as C branch goto label; same as C ement by 1 ** . -same as C same as C The 2020 revision of C++ added a three-way com ator <=> , also known as "the sp hip".

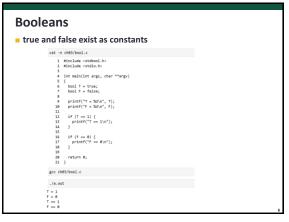
3

1

Booleans

warning due to narrowing conversion
 pre control for a second second

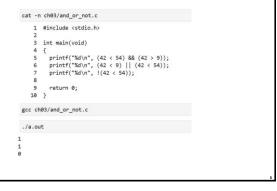
Booleans In c, true reserved by 1 and false by 0 In the intervence of the inter



Boolean Operators

- logical AND: &&
- logical OR: ||
- logical NOT: !
- be sure to use && and ||, as & and | are bitwise operators

Boolean Operators



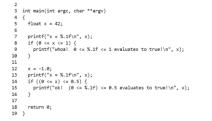
8

7

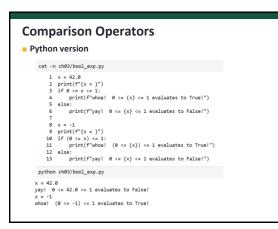
Comparison Operators

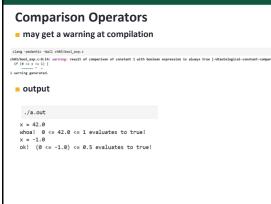
can't use a <= x <= b in C</p>

inte	rpreted as (a <=	x) <= b, where	the first part ret	urns 0 or 1
cat -	ch03/bool_exp.c			
1	<pre>#include <stdio.h></stdio.h></pre>			

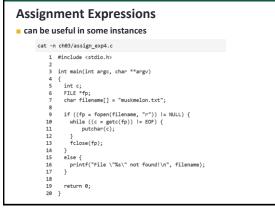


9

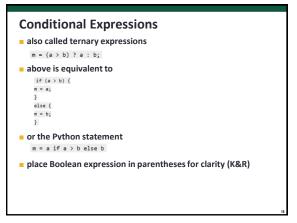




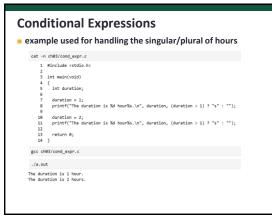
ssi	gnment Expressions
be c	areful with == and =
cat -n	ch03/assign_exp.c
2 3 4 5 6 7 8 9	<pre>Michale cutto.ho { fants.ret dip fa</pre>
clang ·	pedantic -Wall ch03/assign_exp.c
if (x ch03/ass if (x	ign_cost.citis: wrining: using the result of an assignment as a condition without parentheses [-iparentheses] = $\frac{1}{2}$ (ign_cost.citis: note: place parentheses around the assignment to silence this warning = s() { $- \frac{1}{2}$
if (x	ign_exp.c:B:9: note: use 'mm' to turn this assignment into an equality comparison = \$4) { = = g enverted.
./a.out	
x = 42.4 yikes!	

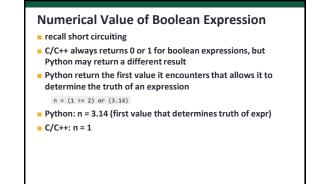


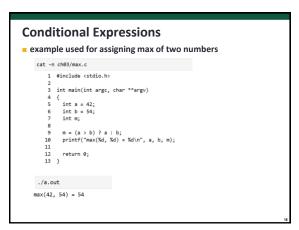


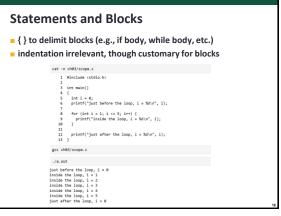




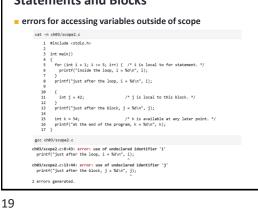


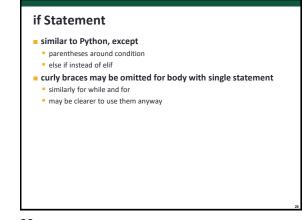


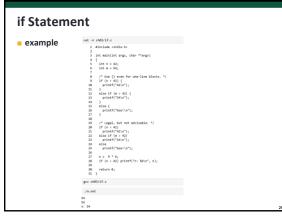


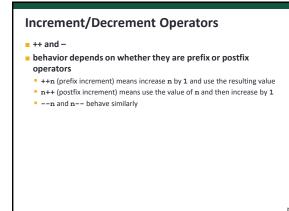


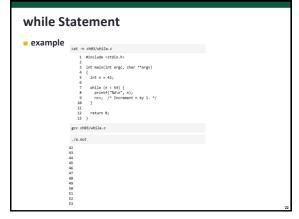
Statements and Blocks

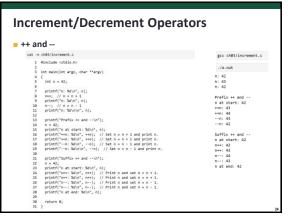












for Statement

- more general than for in Python
- equivalent to while statement
 no iterable objects in C

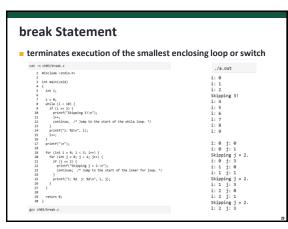
25

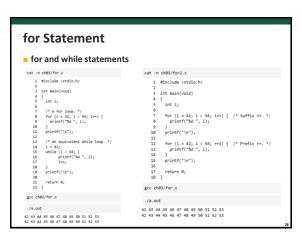
for Statement

common to have temporary local loop variable

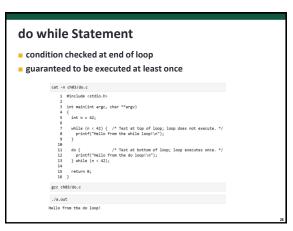
cat -	n ch03/for3.c
1	#include <stdio.h></stdio.h>
2	
3	int main()
4	{
5	int i = 0;
6	printf("Just before the loop, i = %d.\n", i);
7	
8	<pre>for (int i = 1; i <= 5; i++) { /* The i declared here is local to the loop. */ printf("Inside the loop, i = %d.\n", i);</pre>
10	<pre>printr(inside the loop, i = set (n , i); }</pre>
11	,
12	printf("Just after the loop, i = %d.\n", i);
13	}
gee ei	₩83/for3.c
./a.o	it.
ust be	fore the loop, i = 0.
nside	the loop, i = 1.
	the loop, i = 2.
	the loop, i = 3.
	the loop, i = 4.
	the loop, i = 5.
ust af	ter the loop, i = 0.

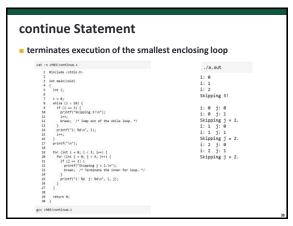
27





26





switch Statement

- no analog in Python
- multiway decision statement
- checks value against constant int expressions and branches accordingly
- executed as follows
 - control expression is evaluated
 - if value matches a case label, the program jumps to the case block
 - If the value is not a match for case label, the default block is executed
 - if no default case, no statements will be executed

switch Statement cat -n ch03/switch.c 1 #include cstdio.h> gcc ch03/switch.c ./a.out int main(void)
{
 int n; ** *** **** Done with first switch statement! , and the set of the 11 12 13 14 15 16 17 18 19 10 Done with second switch statement! No stars for you! Done with third switch statement! n = 7; switch (n) { case 1: printf("* "); case 2: printf("** "); case 4: printf("*** "); case 4: printf("**** "); Done with fourth switch statement!) printf(*\nDone with second switch statement!\n\n"); } printf("\nDone with third switch statement n = 2; smitch (n) { case 2: printf("** "); case 1: printf("**"); case 4: printf("***"); case 4: printf("****"); } printf("\nDone with fourth switch state return #;

32

31

switch Statement

 all cases executed due to lack of break statements between cases



33

