

CS 161: Introduction to Programming and Problem-solving

Warren Harrison

Control Flow and Looping

PORTLAND STATE
UNIVERSITY

Control Flow

- Execution sequence of instructions
- Types of control flow
 - Sequence
 - Looping
 - Choice
- Decision-making

PORTLAND STATE
UNIVERSITY

Doing the Same Set of Instructions Over and Over

```
salesTot = 0
salesAmt = int(input("Enter Store Sale "))
salesTot = salesTot + salesAmt
salesAmt = int(input("Enter Store Sale "))
salesTot = salesTot + salesAmt
salesAmt = int(input("Enter Store Sale "))
salesTot = salesTot + salesAmt
salesAmt = int(input("Enter Store Sale "))
salesTot = salesTot + salesAmt
print("Total Sales:", salesTot)
```

3

PORTLAND STATE
UNIVERSITY

Accumulation

- common activity when writing programs
- often used in summations of values or counting
- **counter variable** – initialized to a starting value
- **increment (or decrement) statement** adds (or subtracts) a particular value to the counter variable: `counter = counter + 1`

4

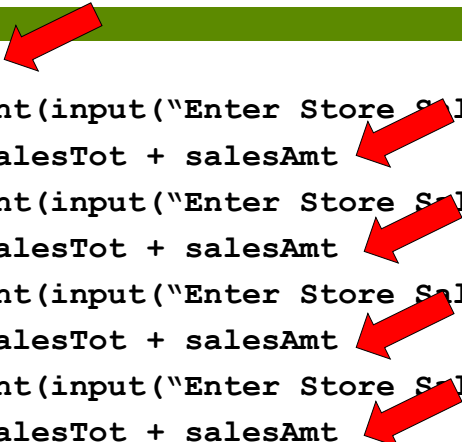
PORTLAND STATE
UNIVERSITY

Doing the Same Set of Instructions Over and Over

```

salesTot = 0
salesAmt = int(input("Enter Store Sale "))
salesTot = salesTot + salesAmt
salesAmt = int(input("Enter Store Sale "))
salesTot = salesTot + salesAmt
salesAmt = int(input("Enter Store Sale "))
salesTot = salesTot + salesAmt
salesAmt = int(input("Enter Store Sale "))
salesTot = salesTot + salesAmt
print("Total Sales:", salesTot)

```



5



Limitations

- Can only collect sales from exactly four stores – not three and not five – ‘Hard coded’
- Write extra code
 - tedious
 - easier to make a mistake
 - if you made a mistake, you might have to change all occurrences of the mistake – perhaps in every statement

6



Looping

- Causes a statement or set of statements to be repeatedly executed:

```
salesAmt = int(input("Enter Store Sale "))  
salesTot = salesTot + salesAmt
```

7

The logo for Portland State University, featuring the text "PORTLAND STATE UNIVERSITY" in a serif font, with "PORTLAND STATE" on the top line and "UNIVERSITY" on the bottom line, all in white text on a dark green rectangular background.

The While Loop

```
storeCount=0  
salesTot = 0  
while(storeCount < 4):  
    salesAmt=int(input("Enter Store Sale "))  
    salesTot = salesTot + salesAmt  
    storeCount = storeCount + 1  
print("Total Sales:",salesTot)
```

8

The logo for Portland State University, featuring the text "PORTLAND STATE UNIVERSITY" in a serif font, with "PORTLAND STATE" on the top line and "UNIVERSITY" on the bottom line, all in white text on a dark green rectangular background.


The While Loop

continuation condition

```

storeCount=0
salesTot = 0
while(storeCount < 4):
    salesAmt=int(input("Enter Store Sale "))
    salesTot = salesTot + salesAmt
    storeCount = storeCount + 1
print("Total Sales:",salesTot)

```



9

PORTLAND STATE
UNIVERSITY

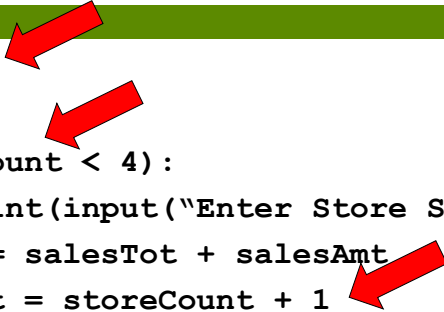
The While Loop

sentry variable

```

storeCount=0
salesTot = 0
while(storeCount < 4):
    salesAmt=int(input("Enter Store Sale "))
    salesTot = salesTot + salesAmt
    storeCount = storeCount + 1
print("Total Sales:",salesTot)

```



10

PORTLAND STATE
UNIVERSITY

Continuation Condition

- The loop continues looping while the condition is true – what conditions?
 - `a < b` less than
 - `a > b` greater than
 - `a == b` equal in value
 - `a != b` not equal in value
- *It is very important that the condition will eventually become false ...*

11


 PORTLAND STATE
UNIVERSITY

Relational Operators

- A Relational Operator asserts that some relationship holds between two items
- `<`, `<=`, `>`, `>=`, `==`, `!=`
- The assertion may be TRUE or FALSE
- `a = 4, b = 3, c = 12, d = 4`
 - `a < b`
 - `a == d`
 - `c != d`
 - `a >= b`

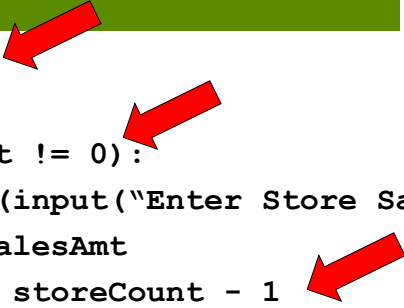
12


 PORTLAND STATE
UNIVERSITY

The While Loop

... but counting down

```
storeCount = 4
salesTot = 0
while(storeCount != 0):
    salesAmt=int(input("Enter Store Sale "))
    salesTot = salesAmt
    storeCount = storeCount - 1
print("Total Sales:",salesTot)
```



13

PORTLAND STATE
UNIVERSITY

Looping a Defined Number of Times

```
counter=0
while(counter < limit):
    <loop body>
    counter = counter + increment
```

14

PORTLAND STATE
UNIVERSITY

Using a Code Tracing Chart

- Make each variable a column head
- Make a column for each condition
- Make a column for output

storeCount	salesTot	salesAmt	storeCount != 0

15


 PORTLAND STATE
UNIVERSITY

Tracing Your Code

- Walk through code one line at a time
- Each time a variable changes, change it on the chart
- Each time you get to a condition statement, evaluate the condition
- For conditions, write TRUE or FALSE

16


 PORTLAND STATE
UNIVERSITY

Looping Until Something Happens

```
totalSales = 0

storeName = input("Enter Store Name ")
while(storeName != "QUIT"):
    storeSales = int(input("enter sales"))
    totalSales = totalSales + storeSales
    storeName = input("Enter Store Name ")

print("Total Sales: ",totalSales)
```

17



Loop Until Match

- Doesn't Use a Counter
- Waits until there is a "match"

```
while(match not found):
    <loop body>
```

- Need to make sure that a match **can** occur

18



The Match Will Never Happen Why?

```
totalSales = 0

storeName = input("Enter Store Name ")
while(storeName != "QUIT"):
    storeSales = int(input("enter sales"))
    totalSales = totalSales + storeSales

print("Total Sales: ",totalSales)
```

19



Relational Operators and Strings

- Relationships between strings are based on the ***ASCII collating sequence***.
- Usually makes sense
 - “a” < “b”
 - “adam” < “andrew”
 - “2” < “3”
 - “1000” < “2” *What?!?!?!?*

20



ASCII Codes

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

Source: www.asciitable.com

21

Sales Report

```
totalSales = storeCount = 0
storeName = input("Enter Store Name ")
while(storeName != "QUIT"):
    storeSales = int(input("Enter Sales "))
    totalSales = totalSales + storeSales
    storeCount = storeCount + 1
    storeName = input("Enter Store Name ")
avgSales = totalSales/storeCount
print("Total Sales: ",totalSales)
print("Average Sales per Store: ",avgSales)
```

22