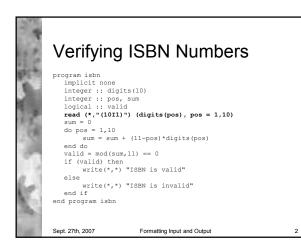
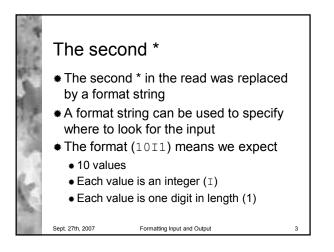
Computers in Engineering COMP 208

Formatting Input and Output Michael A. Hawker





FORTRAN Formats

Sept. 27th, 2007

- The READ and WRITE statements we have seen so far are called *free-format* statements.
- They are easy to use but we have no control over the placement of the input or appearance of the output.
- To control the appearance of the input and output, Fortran allows us to use format specifications

Formatting Input and Output

How much was that? PROGRAM cost IMPLICIT NONE REAL :: price, gst, pst READ(*,*) price gst = 0.075*(price + gst) WRITE(*,*) "Price: ",price WRITE(*,*) "SST: ", gst WRITE(*,*) "Total Cost: ",price+gst+pst END PROGRAM cost

No. or a	The results aren't very pretty.				
8	136.95				
17	Price	: 136.9500			
204	GST:	9.586500			
100	PST:	10.99024			
	Total	Cost: 157.5267			
X					
Sale 1	Sept. 27th, 2007	Formatting Input and Output	6		

「中小い	Wouldn't this be nicer?				
5	136.95				
10	Price:	136.95			
.0	GST:	9.59			
100	PST:	10.99			
1.1	Total Cost:	157.53			
N.					
The gall	Sept. 27th, 2007 Formatting Inp	ut and Output	7		



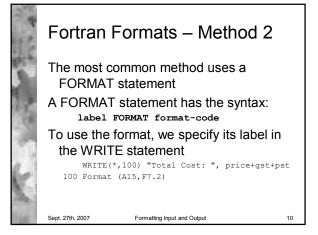
Formats

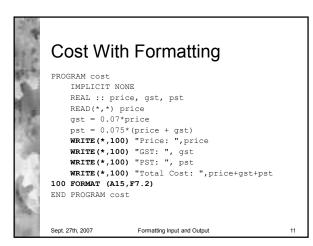
- * FORTRAN formats allow us to specify the placement of values both in output and input
- * Using format descriptors we can control the appearance of output values
- Format descriptors specify
 - The appearance of output values
 - Repetition
 - Vertical positioning
 - Horizontal positioning

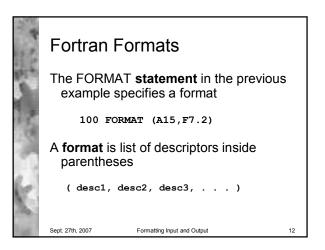
Sept. 27th, 2007 Formatting Input and Output

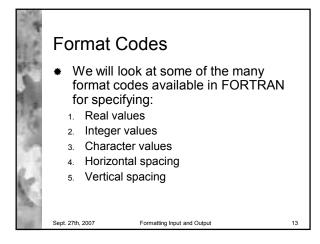
Fortran Formats – Method 1 There are two possible ways to specify a format. In the first, we write the format as a character string and use it to replace the second asterisk in WRITE(*,*). WRITE(*,"(A15,F7.2)") "Total Cost: ", & price+gst+pst Sept. 27th, 2007

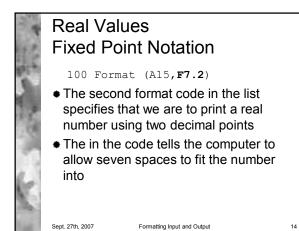
Formatting Input and Output

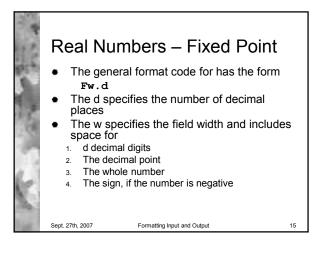


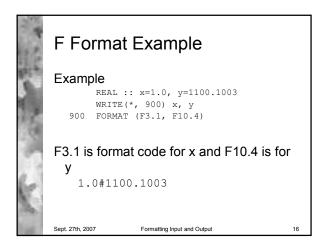


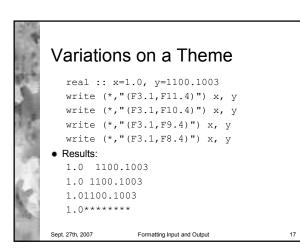










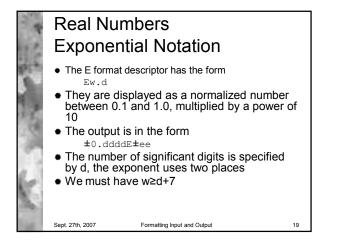


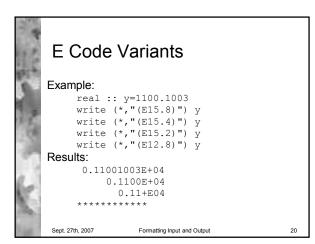
Oops!

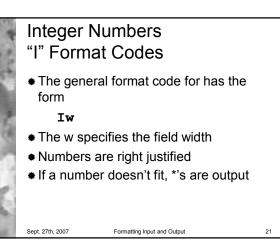
Sept. 27th, 2007

- * What happened in the last example?
- Whenever a value to be output does not fit into the allocated field width, w, the computer just outputs w *'s\
- This is true of any type of value, not just real numbers

Formatting Input and Output







Character Values "A" Format Code

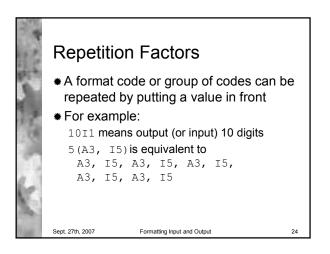
- The general format code for has the form
 Aw
- The w specifies the field width
- Strings are right justified

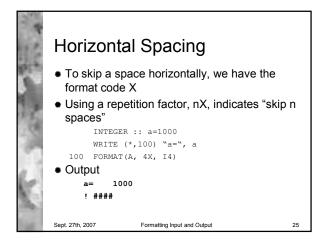
Sept. 27th, 2007

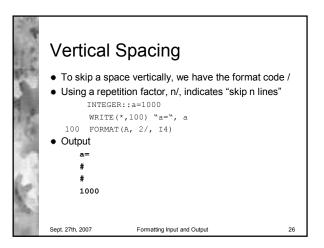
- If a number doesn't fit, the first w characters are output
- If w is left out, the entire character string is printed

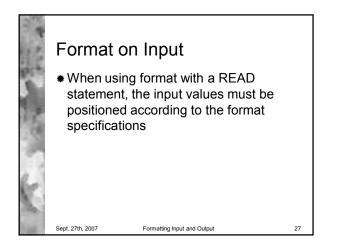
Formatting Input and Output

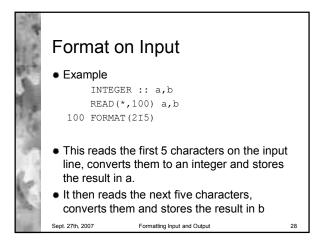
-	Cost With Formatting			
the the C	WRITE(*,100) WRITE(*,100)	, gst, pst ce cice (price + gst) "Price: ",price "GST: ", gst		
R. J.	100 FORMAT (A15, H	7.2)		
1	END PROGRAM cost			
Same?	Sept. 27th, 2007	Formatting Input and Output	23	

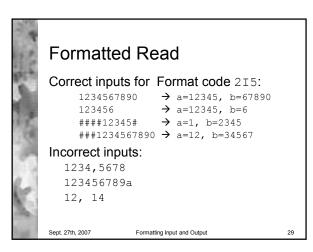












「ちち	Reading Fixed Point Reals				
	Example				
	READ(*,	"(F5.1)") x			
15	Results				
	##3.4	→ x=3.4			
1	123.456	→ x=123.4			
5	12345	→ x=1234.5			
S And	Sept. 27th, 2007	Formatting Input and Output	30		

