Conditionals

Victor Eijkhout, Susan Lindsey

Fall 2022

last formatted: August 28, 2022

1. Conditionals

Single line conditional:

```
if ( test ) statement
```

The full if-statement is:

```
if ( something ) then
  !! something_doing
else
  !! otherwise_else
end if
```

The 'else' part is optional; you can nest conditionals.

2. Comparison and logical operators

Operator	old style	meaning	example
==	.eq.	equals	x==y-1
/=	.ne.	not equals	x*x/=5
>	.gt.	greater	y>x-1
>=	.ge.	greater or equal	sqrt(y)>=7
<	.lt.	less than	
<=	.le.	less or equal	
	.and. .or.	and, or	x<1 .and. x>0
	.not.	not	.not.(x>1 .and. x<2)
	.eqv.	equiv (iff, not XOR)	
	.neqv.	not equiv (XOR)	

3. Select statement

Test single values or ranges, integers or characters:

```
Select Case (i)
Case (:-1)
   print *,"Negative"
Case (5)
   print *,"Five!"
Case (0)
   print *,"Zero."
Case (1:4,6:) ! can not have (1:)
   print *,"Positive"
end Select
```

Compiler does checking on overlapping cases!

Case values need to be constant expressions.

Exercise 1

Read in a positive integer. If it's a multiple of three print 'Fizz!'; if it's a multiple of five print 'Buzz!'. It it is a multiple of both three and five print 'Fizzbuzz!'. Otherwise print nothing.

Note:

- Capitalization.
- Exclamation mark.
- Your program should display at most one line of output.

Optional exercise 2

Read in three grades: Algebra, Biology, Chemistry, each on a scale $1 \cdots 10$. Compute the average grade, with the conditions:

- Algebra is always included.
- Biology is only included if it increases the average.
- Chemistry is only included if it is 6 or more.