

Selection

The Case Statement

# Selection – Making Decisions

We have already used selection for making decisions with the “if-then-else” statements, and its different forms:

if (condition) then ...

if (condition) then...  
else...

if (condition) then...  
elseif (condition2)...  
elseif (condition3)...

# Multiple Selections Using “elsif”

```
if (condition1) then
...
elsif (condition2)
...
elsif (condition3)
...
elsif (condition4)
...
else
...
end if
```

The elsif statement allows multiple conditions to be checked. As soon as one condition is true, the if-elsif statement is finished.

The **else** statement is optional.

# Multiple Selections using “case”

There is another type of selection statement called case. It does not check conditions. Instead, it checks the actual values of a **variable**.

```
case variable of
  label value1 :
    do_something1
  label value2, value3 :
    do_something_else
  label :
    do_something_by_default
end case
```

# Multiple Selections using “case”

```
case variable of  
  label value1 :  
    do_something1  
  label value2, value3 :  
    do_something_else  
  label :  
    do_something_by_default  
end case
```

Notice that the last label has no value. This is called the default case, and contains the code that runs if none of the other values are matched.

# Case Statement – Turing Example

```
case mark of
  label 9, 10 :
    put "Excellent"
  label 7, 8 :
    put "Good"
  label 6 :
    put "Fair"
  label :
    put "Poor"
end case
```

Notice that we are not comparing values using the  $>$ ,  $<$ ,  $=$  operators.

The case statement matches exact values for the variable.

# Case Statement – Turing Example 2

```
case name of
  label "horse", "cow" :
    put "Farm animal"
  label "tiger", "lion" :
    put "Jungle animal"
  label "cat", "dog" :
    put "Pet"
  label :
    put "Unknown animal"
end case
```