Module 3

Decision Structures and Boolean Logic

Instructor: Jonny C.H Yu

ALL programs can be written using three forms of control:

- Sequential structure (we have learned)

```
[ ]: celsius = float(input("What is the Celsius temperature?"))

[ ]: fahrenheit = (9 / 5) * celsius + 32

[ ]: print("The temperature is ", fahrenheit, " degrees Fahrenheit.")

[ ]: print("The temperature is ", format(fahrenheit, '.2f'), " degrees Fahrenheit.")
```

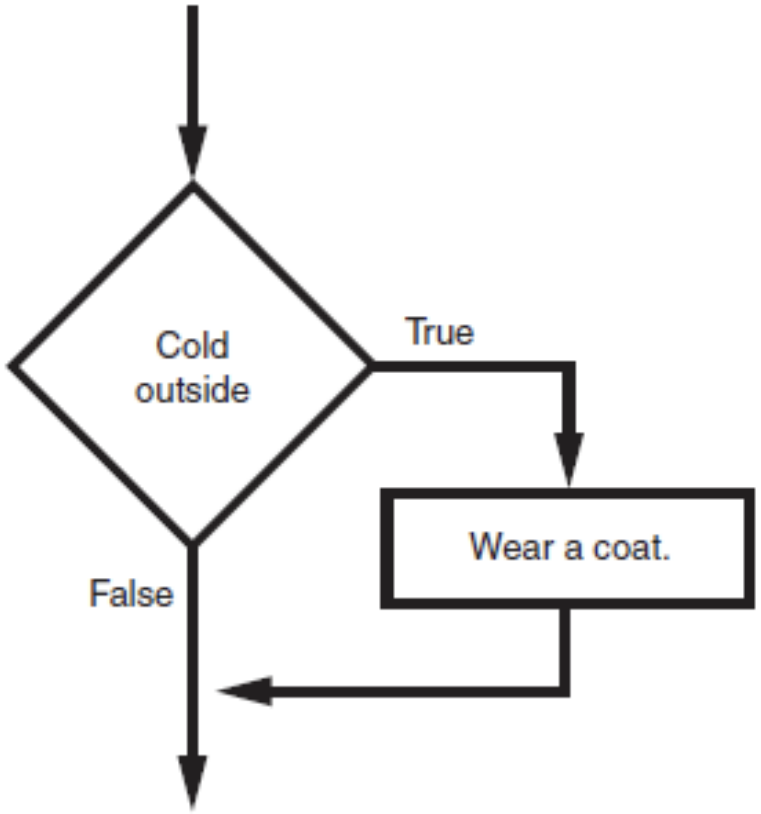
- Decision structure (aka, Selection structure) (this module)
- Repetition structure (next module)

Bohm, C. and G. Jacopini (1966). "Flow Diagrams, Turning Machines and Languages with Only Two Formation Rules, Communications of ACM, 9(5), 366-371.

Decision Structure

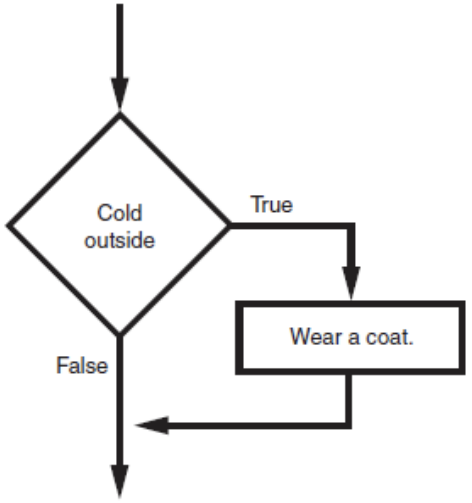
if a <b:

The `if` Statement



- **In flowchart, diamond represents true/false condition that must be tested.**
- **Actions can be *conditionally executed.***
  - Performed only when a condition is true.
- **<u>Single alternative decision structure</u>: provides only one alternative path of execution**
  - If condition is not true, exit the structure

The `if` Statement (cont'd.)



- **Python syntax:**

```
if condition:
    Statement
    Statement
```

- First line known as the `if` clause.
  - Includes the keyword `if` followed by condition.
    - The condition can be true or false.
    - When the `if` statement executes, the condition is tested, and if it is true the block statements are executed. Otherwise, block statements are skipped.

Boolean Expressions and Relational Operators

Boolean expression: expression tested by if statement to determine if it is true or false

- Example: a > b
- `true` if a is greater than b; `false` otherwise

Relational operator: determines whether a specific relationship exists between two values

- Example: greater than (>)

Examples

- Ask the user to enter current temperature in C and give a cold warning when it is below 15 C.

```
[1]: #
     # An if Example: Cold Weather Warning
     #
```

```
[2]: celsius = float(input("What is the Celsius temperature?"))
```

```
What is the Celsius temperature? 14.5
```

```
[3]: if celsius < 15:
         print ("A cold warning")
```

```
A cold warning
```

Let us try it! Download Codes_Module04.zip, upzip and run M4_ColdWarning.ipynb
What happens if you enter 14 for the temperature?
What happens if you enter 15 for the temperature?

Exercise

- Ask the user to enter three testing scores (0-100). Find the average. Print the average up to two significant digits and congratulate the user if the average is equal or higher than 90.

Program Output (with input shown in **bold**)

Enter the first score: **90**
Enter the second score: **92**
Enter the third score: **94.5**
The average score is 92.17
Congratulations!
This is a great average.

Program Output (with input shown in **bold**)

Enter the first score: **90**
Enter the second score: **85**
Enter the third score: **92**
The average score is 89.00

Exercise (Answer)

- Ask the user to enter three testing scores (0-100). Find the average. Print the average up to two significant digits and congratulate the user if the average is equal or higher than 90.

```
[1]: test1 = float(input('Enter the first score: '))
     test2 = float(input('Enter the second score: '))
     test3 = float(input('Enter the third score: '))
```

```
Enter the first score:  90
Enter the second score:  92
Enter the third score:  94.5
```

```
[2]: average = (test1 + test2 + test3) / 3
```

```
[3]: print('The average score is', format(average,'.2f'))
```

```
The average score is 92.17
```

```
[4]: if (average >= 90):
         print('Congratulations!')
         print('This is a great average.')
```

```
Congratulations!
This is a great average.
```

Decision Structure

if a <b:

else:

The `if-else` Statement

- Dual alternative decision structure: two possible paths of execution.
  - One is taken if the condition is true, and the other if the condition is false.

  - Syntax:
```
if condition:
        statements
else:
        other statements
```

  - `if` clause and `else` clause must be aligned
  - Statements must be consistently indented

Exercise

- Ask the user to enter current temperature in C.

If temperature ≧15 C, print “Nice weather we are having”.
Otherwise print “A little cold, isn’t it?”

```
[ ]: #
     # An if-else Example: Nice or Cold
     #
```

```
[ ]: celsius = float(input("What is the Celsius temperature?"))
```

```
[ ]: if celsius >= 15:
         print ("Nice weather we are having")
     else:
         print ("A little cold, isn’t it?")
```

Let us try it! (M4_ColdNice.ipynb)

1. **What happens if you enter 14 for the temperature?**
2. **What happens if you enter 15 for the temperature?**

Exercise

- Ask the user to enter three testing scores (0-100). Find the average. Print the average up to two significant digits. Congratulate the user if the average is equal or higher than 90; otherwise say some encouraging words.

Program Output (with input shown in **bold**)

Enter the first score: **90**
Enter the second score: **92**
Enter the third score: **94.5**
The average score is 92.17
Congratulations!
This is a great average.

Program Output (with input shown in **bold**)

Enter the first score: **90**
Enter the second score: **85**
Enter the third score: **92**
The average score is 89.00
Keep going!
You can make it.

Exercise (Answer)

- Ask the user to enter three testing scores (0-100). Find the average. Print the average up to two significant digits and congratulate the user if the average is equal or higher than 90. otherwise say some encouraging words.

```
[ ]: test1 = float(input('Enter the first score: '))
     test2 = float(input('Enter the second score: '))
     test3 = float(input('Enter the third score: '))
```

```
[ ]: average = (test1 + test2 + test3) / 3
```

```
[ ]: print('The average score is', format(average,'.2f'))
```

```
[ ]: if (average >= 90):
         print('Congratulations!')
         print('This is a great average.')
     else:
         print('Kep going!')
         print('You can make it.')
```

The `if-elif-else` Statement

- `if-elif-else` statement: special version of a decision structure.
  - Makes logic of nested decision structures simpler to write
    - Can include multiple `elif` statements
  - Syntax:

```
if condition_1:
    statement(s)
elif condition_2:
    statement(s)
elif condition_3:
    statement(s)
else:
    statement(s)
```

Insert as many `elif` clauses as necessary.

The `if-elif-else` Statement (cont'd)

- Alignment used with `if-elif-else` statement:
  - `if`, `elif`, and `else` clauses are all aligned
  - Conditionally executed blocks are consistently indented
- `if-elif-else` statement is never required, but logic easier to follow
  - Can be accomplished by nested `if-else`
    - Code can become complex, and indentation can cause problematic long lines

Exercise

- NCKU grading system maps your score in number grade (百分數) to that in letter grade (等第). Write a program that allows the user to input a score and display a letter grade. As an exercise for simplicity, let us treat the score below 77 as an invalid input.

百分數		等第
90-100	95	A+
85-89	87	A
80-84	82	A-
77-79	78	B+
73-76	75	B
70-72	70	B-
67-69	68	C+
63-66	65	C
60-62	60	C-
≦ 59	50	F
0	0	X

Program Output (with input shown in **bold**)

Enter a score: **83.2**

The score 83.2 is A-

Program Output (with input shown in **bold**)

Enter a score: **75**

The score 75.0 is an invalid input

Exercise (Answer)

- NCKU grading system maps your score in number grade (百分數) to that in letter grade (等第). Write a program that allows the user to input a score and display a letter grade. As an exercise for simplicity, let us treat the score below 77 as an invalid input.

```
[ ]: score = float(input('Enter a score:'))
```

```
[ ]: if (score >= 90):
         grade = 'A+'
     elif (score >= 85):
         grade = 'A'
     elif (score >= 80):
         grade = 'A-'
     elif (score >= 77):
         grade = 'B+'
     else:
         grade = 'an invalid input'
```

```
[ ]: print('The score', score, 'is', grade)
```

百分數		等第
90-100	95	A+
85-89	87	A
80-84	82	A-
77-79	78	B+
73-76	75	B
70-72	70	B-
67-69	68	C+
63-66	65	C
60-62	60	C-
≦ 59	50	F
0	0	X

Logical Operators

- Logical operators: operators that can be used to create complex Boolean expressions.
  - `and` operator and `or` operator: binary operators, connect two Boolean expressions into a compound Boolean expression.
  - `not` operator: unary operator, reverses the truth of its Boolean operand.

Table 3-4 Compound Boolean expressions using logical operators

Expression	Meaning
`x > y and a < b`	Is x greater than y AND is a less than b?
`x == y or x == z`	Is x equal to y OR is x equal to z?
`not (x > y)`	Is the expression x > y NOT true?

Exercise

- A bank will grant the loan if one earns 500,000 per year and is employed for 2 years. Write a program to determine whether the user qualifies for a loan.

```
[ ]: MIN_SALARY = 500000
     MIN_YEARS = 2
```

```
[ ]: salary = float(input('Enter your annual salary:'))
```

```
[ ]: years_on_job = float(input('Enter the number of years employed:'))
```

```
[ ]: if salary > MIN_SALARY and years_on_job > MIN_YEARS:
         print('You quality for the loan.')
     else:
         print('You do not quality for the loan.')
```

M4_LoanQualifier.ipynb

Boolean Variables

- Boolean variable: references one of two values, `True` or `False`
  - Represented by `bool` data type

- Commonly used as flags
  - Flag: variable that signals when some condition exists in a program
    - Flag set to `False` → condition does not exist
    - Flag set to `True` → condition exists

Summary

- This module covered:
  - Decision structures, including:
    - Single alternative decision structures
    - Dual alternative decision structures
    - Nested decision structures
  - Relational operators and logical operators as used in creating Boolean expressions
  - String comparison as used in creating Boolean expressions
  - Boolean variables

To be continued…