Data Processing and Analysis in Python
Lecture 3
Selection Statements

UNIVERSITY OF MARYLAND

ROBERT H. SMITH SCHOOL OF BUSINESS

DR. ADAM LEE

Boolean Type and Expressions

- **Boolean** data type consists of two values: **True** and **False**

```
>>> type(True)
<type 'bool'>
```

Comparison Operator	Meaning
==	Equals
!=	Not equals
<	Less than
<=	Less than or equal
>	Greater than
>=	Greater than or equal

Selection: One-Way If Statements

- **Selection** statements allow a computer to make choices
  - Based on a **condition** that generates **True/False**

- **if**: single/one-way selection statement
- **if** <condition>**:**
  <sequence of statements – True block>

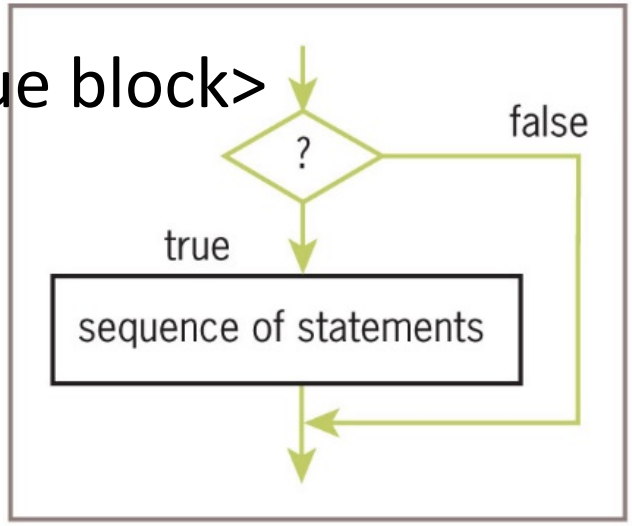

Figure 3-3 The semantics of the `if` statement

Selection: Two-Way If-Else Statements

- **if-else**: double/dual/two-way selection statement
- **if** <condition>**:**

  <sequence of statements – True block>

  else:

  <sequence of statements – False block>

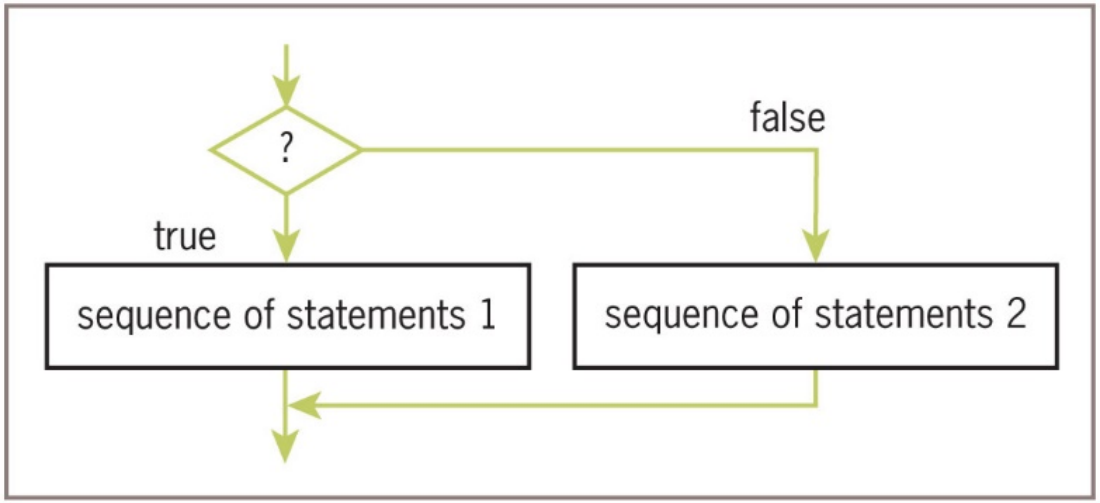

Figure 3-2 The semantics of the `if-else` statement

Selection: Nested If-Else Statements

- A program may be faced with testing conditions that entail more than two alternative courses of action

Letter Grade	Range of Numeric Grades
A	All grades at or above 90
B	All grades at or above 80 and below 90
C	All grades at or above 70 and below 80
F	All grades below 70

Selection: Multi-Way If-Elif-Else Statements

- **if-elif-else**: multiple/multi-way selection statement
- **if** <condition-1>**:**
  <sequence of statements-1>
  elif <condition-*n*>**:**
  <sequence of statements-*n*>
  else:
  <default sequence of statements>

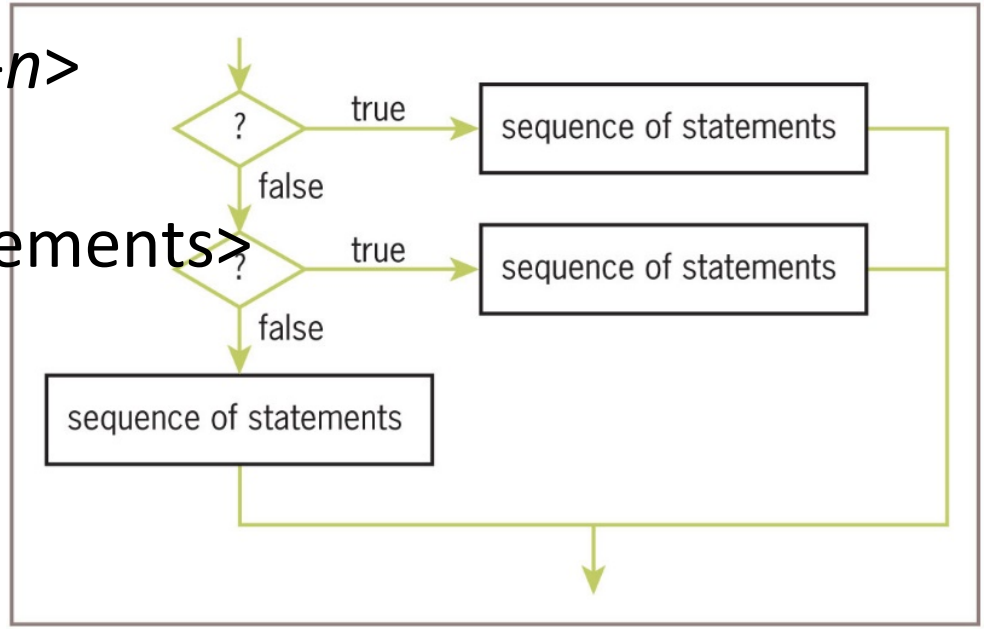

Figure 3-4 The semantics of the multi-way `if` statement

Compound Boolean Expressions

- Actions must be taken if either of two conditions is true:

```
number = int(input("Enter the numeric grade: "))
if number > 100:
    print("Error: grade must be between 100 and 0")
elif number < 0:
    print("Error: grade must be between 100 and 0")
else:
     # The code to compute and print the result go here
```

- Simplified Code:

```
number = int(input("Enter the numeric grade: "))
if number > 100 or number < 0:
    print("Error: grade must be between 100 and 0")
else:
    # The code to compute and print the result go here
```

Logical Operators and Truth Tables

A	B	A and B
True	True	True
True	False	False
False	True	False
False	False	False

A	B	A or B
True	True	True
True	False	True
False	True	True
False	False	False

A	not A
True	False
False	True

A	True	False
not A	**False**	**True**

A **and** B	B = True	B = False
A = True	**True**	**False**
A = False	**False**	**False**

A **or** B	B = True	B = False
A = True	**True**	**True**
A = False	**True**	**False**

Figure 3-5 The truth tables for **and**, **or**, and **not**

Precedence of Operators

Type of Operator	Operator Symbol
Parentheses	()
Exponentiation	**
Arithmetic negation	–
Multiplication, division, remainder	*, /, //, %
Addition, subtraction	+, –
Comparison	==, !=, <, >, <=, >=
Logical negation	not
Logical conjunction	and
Logical disjunction	or
Assignment	=

UNIVERSITY OF MARYLAND

Short-Circuit Evaluation

- In (A **and** B), if A is **False**, then so is the expression, and there is no need to evaluate B
- In (A **or** B), if A is **True**, then so is the expression, and there is no need to evaluate B
- Short-circuit: Evaluation stops as soon as possible

```
count = int(input("Enter the count: "))
theSum = int(input("Enter the sum: "))
if count > 0 and theSum // count > 10:
    print("average > 10")
else:
    print("count = 0 or average <= 10")
```

UNIVERSITY OF MARYLAND