# ECE12: Introduction to Programming
# Lecture 6

Rainer Dömer

doemer@uci.edu

The Henry Samueli School of Engineering
Electrical Engineering and Computer Science
University of California, Irvine

# Lecture 6: Overview

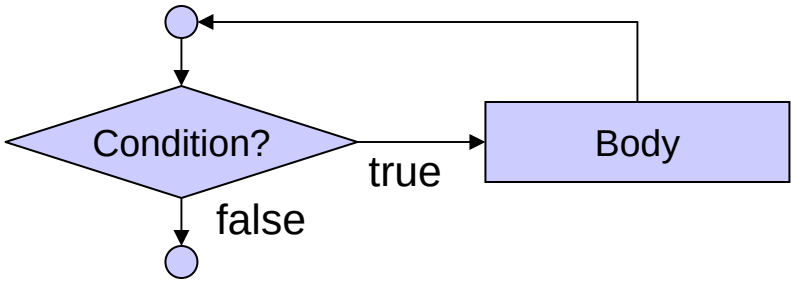
- Augmented Assignment Operations
  - Counter
- Control Structures
  - **`while`** loop
  - **`break`** statement
  - **`continue`** statement
  - **`range`** function
  - **`for`** loop

# Augmented Assignment Operations

- Assignment operator: **=**
  - evaluates right-hand side
  - assigns result to left-hand side
- Augmented assignment operator: **+=**, ***=**, ...
  - evaluates right-hand side as temp. result
  - applies operation to left-hand side and temp. result
  - assigns result to left-hand side
- Example: Counter
  - `x = 0` # initialization
  - `x = x + 1` # counting by regular assignment
  - `x += 1` # counting by augmented assignment
- Augmented assignment operators:
  - **+=**, **-=**, ***=**, **/=**, **%=**, ****=**, **<<=**, **>>=**, **^=**, **|=**, **&=**

# Control Structures

- **`while`** loop
  - Repetition structure (iteration)
  - Flow chart:




  - Example:

```
product = 2
while product <= 1000:
    product *= 2
print product
```

# Example: Average of Numbers

- Version 1: Counter-controlled repetition

(exactly 10 values must be entered)

```
# average.py: compute the average of a set of numbers
#
# author: Rainer Doemer
#
# modifications:
# 01/19/04 RD    initial version

# initialize
count = 0
sum = 0.0

# input and compute
while count < 10:
    x = float(raw_input("Please enter a number: "))
    sum += x
    count += 1

# compute
average = sum / count

# output
print "The sum is", sum
print "The average is", average
```

# Example: Average of Numbers

- Version 2: Sentinel-controlled repetition

(number of values entered is determined by the user at run-time)

```
# average2.py: compute the average of a set of numbers
#
# author: Rainer Doemer
#
# modifications:
# 01/19/04 RD       initial version (based on average.py)

# initialize
count = 0
sum = 0.0

# input
s = raw_input("Enter a number or type 'q' to quit: ")

# compute and input
while s != 'q':
    x = float(s)
    sum += x
    count += 1
    s = raw_input("Enter a number or type 'q' to quit: ")

# compute and output
if count > 0:
    average = sum / count
    print count, "numbers entered."
    print "The sum is", sum
    print "The average is", average
else:
    print "No numbers entered."
```

# Control Structures

- **`break`** statement
  - exits the innermost loop
- **`continue`** statement
  - jump to the beginning of the innermost loop
- Example:

```
i = 0
s = 0
while 1:                # “endless” loop
    i += 1
    if i > 100:
        break;          # exit the loop!
    if i % 2 == 1:
        continue        # next iteration!
    s += i
print s
```

# Example: Average of Numbers

- Version 3: **`break`** statement

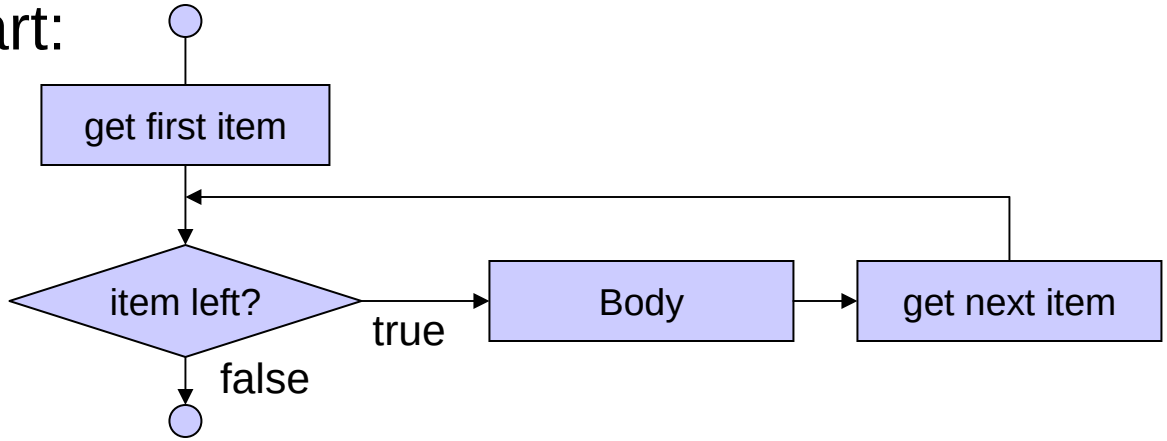
```
# average3.py: compute the average of a set of numbers
#
# author: Rainer Doemer
#
# modifications:
# 01/19/04 RD       initial version (based on average2.py)

# initialize
count = 0
sum = 0.0

# input and compute
while 1:
    s = raw_input("Enter a number or type 'q' to quit: ")
    if s == 'q':
          break;
    x = float(s)
    sum += x
    count += 1

# compute and output
if count > 0:
    average = sum / count
    print count, "numbers entered."
    print "The sum is", sum
    print "The average is", average
else:
    print "No numbers entered."
```

# Control Structures

- **`range()`** function
  - returns a list of values in the range specified as argument
  - one argument: **`range(end)`**
    - returns list of values from **`0`** to **`(end-1)`**
    - Example: **`range(5)`** returns **`[0, 1, 2, 3, 4]`**
  - two arguments: **`range(start, end)`**
    - returns list of values from **`start`** to **`(end-1)`**
    - Example: **`range(2, 6`**) returns **`[2, 3, 4, 5]`**
  - three arguments: **`range(start, end, increment)`**
    - returns list of values from **`start`** to **`(end-increment)`** in steps of **`increment`**
    - Example: **`range(6, 2, -1)`** returns **`[6, 5, 4, 3]`**

# Control Structures

- **`for`** loop
  - Repetition structure (iteration over lists)
  - Flow chart:




  - Example:

```
for name in [“Alan”, “Bob”, “Charlie”]:
    print name
for i in range(1,10):
    print i
```

# Example: Compound Interest

- Example:

```
# interest.py: compute compound interest
#
# author: Rainer Doemer
#
# modifications:
# 01/19/04 RD  initial version

# input
amount = float(raw_input("Enter the principal: "))
apr = float(raw_input("Enter the interest rate: "))

# compute and output
for year in range(1,11):
    amount += amount * (apr/100.0)
    print "End of year %2d: amount on deposit = %8.2f" \
                % (year,amount)
```