



ECE12: Introduction to Programming

Lecture 7

Rainer Dömer

doemer@uci.edu

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

Lecture 7: Overview

- Functions
 - Introduction
 - Concepts
 - Types of functions
 - function call, arguments
 - function parameters, return values
 - Examples
 - square()
 - interest()
 - Module math Functions
 - Example

Functions

- Introduction
 - Programming concepts
 - Hierarchy
 - Encapsulation
 - Information hiding
 - Divide and conquer
 - Software reuse
 - Don't re-invent the wheel!
 - Program components
 - Functions
 - Classes
 - Modules, packages
 - Libraries

Functions

- Types
 - Programmer-defined functions
 - Library functions
- Concepts
 - Function call
 - caller invokes a function
 - Function arguments
 - arguments supply data to the function
 - Function parameters
 - input data supplied to the function
 - Return value
 - output data computed by the function
 - Local variables

Functions

- Example

- $y = \text{square}(x)$

```
# function definition
def square(x):
    y = x * x
    return y

# function call
print square(8.0)
```

- Function definition
 - function name: square
 - function parameter: x
 - function return value: y
 - Function call
 - argument: 8.0
 - result: 64.0

Example: Compound Interest

- Original version from previous lecture

```
# 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)
```

Example: Compound Interest

- New version with `interest()` function

```
# interest2.py: compute compound interest
# author: Rainer Doemer
# modifications:
# 01/26/04 RD  modified for demonstration of functions
# 01/19/04 RD  initial version

# function definition
def interest(principal, rate):
    return principal * (rate/100.0)

# 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 += interest(amount, apr)
    print "End of year %2d: amount on deposit = %8.2f" \
          % (year, amount)
```

Module `math` Functions

- Math module
 - part of Python standard library
 - standard mathematical functions
- Functions provided by `math`
 - `acos()`
 - `asin()`
 - `atan()`
 - `ceil()`
 - `cos()`
 - `exp()`
 - `fabs()`
 - `floor()`
 - `fmod()`
 - `hypot()`
 - `log()`
 - `log10()`
 - `pow()`
 - `sin()`
 - `sqrt()`
 - `tan()`
 - ...
- Importing functions from the `math` module
 - Example
 - `import math`
 - `print math.sqrt(9.0)`

Example: Function Table

```
# function.py: compute a function table

# import modules
import math

# function definition
def f(x):
    return math.cos(x)

# input
lo = float(raw_input("Please enter lower bound: "))
hi = float(raw_input("Please enter upper bound: "))

# compute and output
step = (hi - lo) / 10
x = lo
while x <= hi:
    y = f(x)
    print "f(%12.6f) = %12.6f" % (x,y)
    x += step
```