Outline

1. final exam on Tuesday 3 May 2016, at 8AM
   - general information
   - policies for the final exam
2. some example questions
   - strings, lists, dictionaries
   - scope of variables in functions
   - working with files
   - list comprehensions
   - exception handling
   - GUI design

MCS 260 Lecture 43
Introduction to Computer Science
Jan Verschelde, 27 April 2016

review of Python programming

1. final exam on Tuesday 3 May 2016, at 8AM
   - general information
   - policies for the final exam
2. some example questions
   - strings, lists, dictionaries
   - scope of variables in functions
   - working with files
   - list comprehensions
   - exception handling
   - GUI design

general information

The exam will take place on Tuesday 3 May, from 8AM till 10AM.

Bring your own paper to write the solutions.

If an emergency prevents you from participation,
please contact me as soon as you are able to
so we can schedule a makeup exam in finals week.

The final exam is comprehensive and covers the entire course.

Please review the posted answers to the midterm and the quizzes.

review of Python programming

1. final exam on Tuesday 3 May 2016, at 8AM
   - general information
   - policies for the final exam

2. some example questions
   - strings, lists, dictionaries
   - scope of variables in functions
   - working with files
   - list comprehensions
   - exception handling
   - GUI design

policies for the final exam

The material in this part has its focus on Python.

The exam will be closed book, no notes, and no computer.

Some of the types of questions you could expect:

- Translate pseudo code or flow chart into Python code.
- Execute Python code and write computed values.
- Define a function based on its specification.
- Give code for a method in a class.

This review contains some preliminary examples of questions which may help you prepare for the second part of the midterm exam.

review of Python programming

1. final exam on Tuesday 3 May 2016, at 8AM
   - general information
   - policies for the final exam

2. some example questions
   - strings, lists, dictionaries
   - scope of variables in functions
   - working with files
   - list comprehensions
   - exception handling
   - GUI design

strings, lists, dictionaries

The most important composite data structures in Python are strings, lists, and dictionaries.

Convert the string `"123.45 euro"` into `"135.80 dollar"`.

1. Use a dictionary to store the exchange rates:
   - one euro is 1.10 dollar;
   - one dollar is 0.91 euro.
2. Write Python code to extract the number and currency type from strings such as `"123.45 euro"` and `"135.80 dollar"`.
3. Use the dictionary to make the string that represents the converted amount of money.

review of Python programming

1. final exam on Tuesday 3 May 2016, at 8AM
   - general information
   - policies for the final exam

2. some example questions
   - strings, lists, dictionaries
   - scope of variables in functions
   - working with files
   - list comprehensions
   - exception handling
   - GUI design

scope of variables in functions

Consider the code below:

```
(a, b) = (2, 3)
def update(u, v):
    r = u + v
    return r
a = update(a, b)
print a
```

Answer the following questions:

(1) Which variables are global? _____

(2) Which variables are local? _____

(3) What does the code print? _____

(4) Complete the table below with values for `a`, `b`, `u`, `v`, and `r`.

	`a`	`b`	`u`	`v`	`r`
before the call to `update(a, b)`					
during the call, just before `return r`					
after the call to `update(a, b)`					

review of Python programming

1. final exam on Tuesday 3 May 2016, at 8AM
   - general information
   - policies for the final exam

2. some example questions
   - strings, lists, dictionaries
   - scope of variables in functions
   - working with files
   - list comprehensions
   - exception handling
   - GUI design

working with files

The Unix command `cal` produces a calendar as

```
     April 2016
Su Mo Tu We Th Fr Sa
                1  2
 3  4  5  6  7  8  9
10 11 12 13 14 15 16
17 18 19 20 21 22 23
24 25 26 27 28 29 30
```

Assume that this output of `cal 4 2016` is on file `cal.txt`.
Design an algorithm that, given a number between 1 and 30, returns the day of the week.
For example, if the given number is 2, then `Sa` is returned.
Write Python code for the script.

review of Python programming

1. final exam on Tuesday 3 May 2016, at 8AM
   - general information
   - policies for the final exam

2. some example questions
   - strings, lists, dictionaries
   - scope of variables in functions
   - working with files
   - list comprehensions
   - exception handling
   - GUI design

list comprehensions

Use a list comprehension to compute the coordinates of the points of a regular n-gon.

The points lie on a circle with radius one.

The list on return is a list of tuples with the x- and y-coordinates of the points.

The formula for the coordinates (x,y) is

$$\left(\cos\left(\frac{2k\pi}{n}\right), \sin\left(\frac{2k\pi}{n}\right) \right).$$

more list comprehensions

Take the list of the coordinates of the corner of a regular n-gon and ...

1. Make a list of pairs of consecutive points.
   Each pair in this list spans a line segment.
2. Compute a list of differences of the pairs in the list.
   Now we have a list of vectors.
3. For each line segment, compute its length, and take its sum.

For a large enough number of points, the sum equals 2π.

review of Python programming

1. final exam on Tuesday 3 May 2016, at 8AM
   - general information
   - policies for the final exam

2. some example questions
   - strings, lists, dictionaries
   - scope of variables in functions
   - working with files
   - list comprehensions
   - exception handling
   - GUI design

exception handling

Write a Python function `prompt_integer()`
that returns a user given integer.

The function must ask the user *each time*
to retry when the conversion of the input into an integer fails.

review of Python programming

1. final exam on Tuesday 3 May 2016, at 8AM
   - general information
   - policies for the final exam

2. some example questions
   - strings, lists, dictionaries
   - scope of variables in functions
   - working with files
   - list comprehensions
   - exception handling
   - GUI design

GUI design

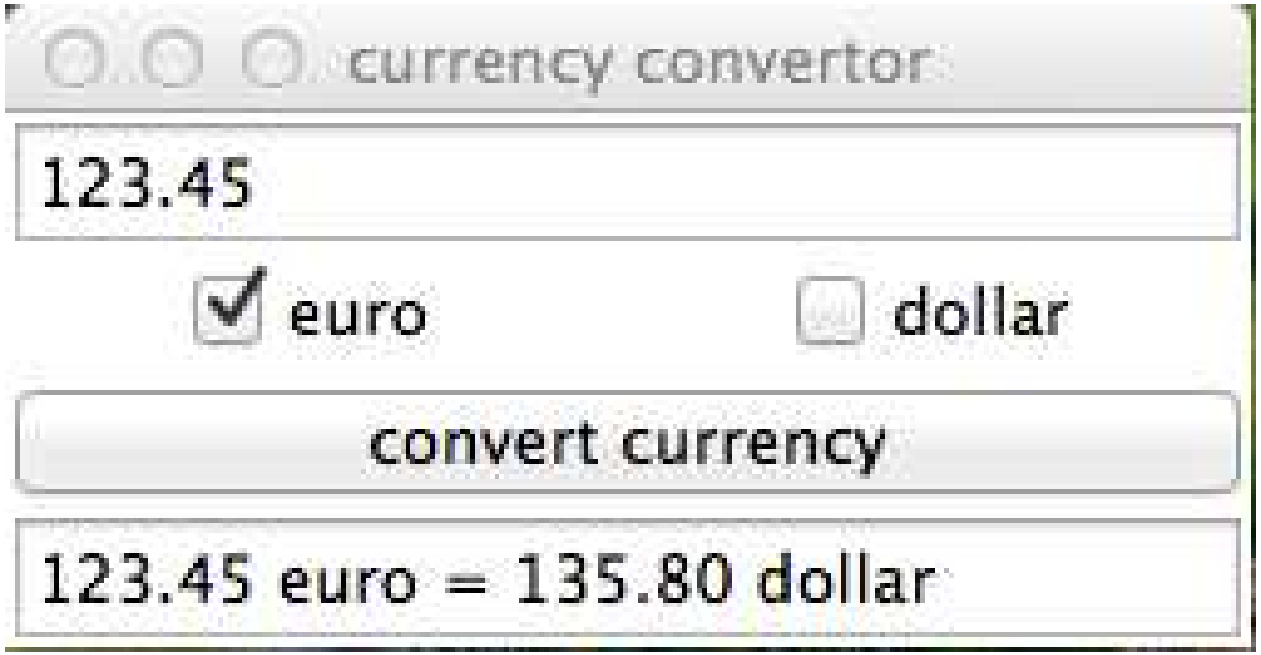
Design a GUI for a currency convertor:

- one euro is worth 1.10 dollar;
- one dollar equals 0.91 euro.

Some types of questions concerning this GUI:

1. Describe a design for this GUI.
   What kind of widgets will you use?
2. Given a picture of the design, name the types of widgets.
3. Given the design, write the constructor in the object oriented implementation to define the layout of the GUI.
4. Given the constructor, with all the data attributes defined, write code for the callback method in the GUI.

a possible solution to the design



- the first entry widget to enter the amount;
- two check buttons to select the type of currency;
- one button to compute the conversion;
- the second entry widget to display the converted amount.

code for the constructor

```
from tkinter import Tk, Entry, Button, Checkbutton, IntVar, W, E
from tkinter import END, INSERT

def __init__(self, wdw):

    wdw.title('currency convertor')
    self.amount = Entry(wdw, width=30)
    self.amount.grid(row=0, columnspan=2)
    self.curry = IntVar() # currency type
    self.euro = Checkbutton(wdw, text="euro", \
        variable=self.curry, onvalue = 1, offvalue = 0)
    self.dollar = Checkbutton(wdw, text="dollar", \
        variable=self.curry, onvalue = 2, offvalue = 0)
    self.euro.grid(row=1, column=0)
    self.dollar.grid(row=1, column=1)
    self.conv = Button(wdw, text='convert currency', \
        command=self.convert)
    self.conv.grid(row=2, columnspan=2, sticky=W+E)
    self.result = Entry(wdw, width=30)
    self.result.grid(row=3, columnspan=3)
```

code for the callback method

```
EURO2DOLLAR = 1.10
DOLLAR2EURO = 0.91

def convert(self):
    """
    Gets the amount from the entry called amount,
    takes the value from the check button,
    convert the amount using the rates,
    and places the converted amount in the result entry.
    """
    amt = float(self.amount.get())
    display = '%.2f' % amt
    conversionType = self.curry.get()
    if(conversionType == 1): # convert euro to dollar
        res = amt*EURO2DOLLAR
        display = display + ' euro = %.2f dollar' % res
    if(conversionType == 2): # convert dollar to euro
        res = amt*DOLLAR2EURO
        display = display + ' dollar = %.2f euro' % res
    self.result.delete(0, END)
    self.result.insert(INSERT, display)
```