

Graphical User Interfaces

characteristics and components

Graphical User Interfaces

characteristics and components

The Tk GUI Toolkit in Python

Tkinter, Tk, and Tcl

hello world again

using grid, Radiobutton, and Checkbutton

The Tk GUI Toolkit in Python

Tkinter, Tk, and Tcl
hello world again
using grid, Radiobutton,
and Checkbutton

Summary + Assignments

Summary + Assignments

MCS 260 Lecture 30
Introduction to Computer Science
Jan Verschelde, 5 November 2007

Graphical User Interfaces characteristics and components

Graphical User Interfaces

characteristics and
components

The Tk GUI Toolkit in Python

Tkinter, Tk, and Tcl

hello world again

using grid, Radiobutton, and Checkbutton

The Tk GUI Toolkit in Python

Tkinter, Tk, and Tcl
hello world again
using grid, Radiobutton,
and Checkbutton

Summary + Assignments

Summary + Assignments

User Interfaces (UI)

GUI, WUI, and HUI

Interfaces to our programs so far were command line.

Consequences:

1. strict linear control of execution order imposed on the user by program;
2. user must remember a lot, e.g.: file names.

Modern interfaces are mostly *event driven* and give the user control over the actions.

Command line interfaces are call non-GUI.

Specific categories of UI are

WUI: *Web* UI and HUI: *Handheld* UI.

Goal: increase *usability* of software.

User Interfaces (UI)

GUI, WUI, and HUI

Interfaces to our programs so far were command line.

Consequences:

1. strict linear control of execution order imposed on the user by program;
2. user must remember a lot, e.g.: file names.

Modern interfaces are mostly *event driven* and give the user control over the actions.

Command line interfaces are call non-GUI.

Specific categories of UI are

WUI: *Web* UI and HUI: *Handheld* UI.

Goal: increase *usability* of software.

User Interfaces (UI)

GUI, WUI, and HUI

Interfaces to our programs so far were command line.

Consequences:

1. strict linear control of execution order imposed on the user by program;
2. user must remember a lot, e.g.: file names.

Modern interfaces are mostly *event driven* and give the user control over the actions.

Command line interfaces are call non-GUI.

Specific categories of UI are

WUI: *Web* UI and HUI: *Handheld* UI.

Goal: increase *usability* of software.

User Interfaces (UI)

GUI, WUI, and HUI

MCS 260 L-30

5 November 2007

Interfaces to our programs so far were command line.

Consequences:

1. strict linear control of execution order imposed on the user by program;
2. user must remember a lot, e.g.: file names.

Modern interfaces are mostly *event driven* and give the user control over the actions.

Command line interfaces are call non-GUI.

Specific categories of UI are

WUI: *Web* UI and HUI: *Handheld* UI.

Goal: increase *usability* of software.

Graphical User Interfaces

characteristics and components

The Tk GUI Toolkit in Python

Tkinter, Tk, and Tcl
hello world again
using grid, Radiobutton,
and Checkbutton

Summary + Assignments

User Interfaces (UI)

GUI, WUI, and HUI

Interfaces to our programs so far were command line.

Consequences:

1. strict linear control of execution order imposed on the user by program;
2. user must remember a lot, e.g.: file names.

Modern interfaces are mostly *event driven* and give the user control over the actions.

Command line interfaces are call non-GUI.

Specific categories of UI are

WUI: *Web* UI and HUI: *Handheld* UI.

Goal: increase *usability* of software.

User Interfaces (UI)

GUI, WUI, and HUI

Interfaces to our programs so far were command line.

Consequences:

1. strict linear control of execution order imposed on the user by program;
2. user must remember a lot, e.g.: file names.

Modern interfaces are mostly *event driven* and give the user control over the actions.

Command line interfaces are call non-GUI.

Specific categories of UI are

WUI: *Web* UI and HUI: *Handheld* UI.

Goal: increase *usability* of software.

Characteristics of GUIs

The most significant features of a GUI are

Window: area of display device to view and interact with an object. Information is *viewed*.

Icon: pictorial representation of an object.

Menu: collection of choices, typically to perform actions on an object.

Pointer: is similar to a typing cursor, controlled by a pointing device, typically a mouse.

Client Area: area within a window where the user can enter data, strings or numbers.

Directness: direct manipulation of objects via the pointer, such as moving and dragging windows.

The development of GUIs should be *user centered*.

Goals: simple, aesthetic, productive, and customizable.

Characteristics of GUIs

The most significant features of a GUI are

Window: area of display device to view and interact with an object. Information is *viewed*.

Icon: pictorial representation of an object.

Menu: collection of choices, typically to perform actions on an object.

Pointer: is similar to a typing cursor, controlled by a pointing device, typically a mouse.

Client Area: area within a window where the user can enter data, strings or numbers.

Directness: direct manipulation of objects via the pointer, such as moving and dragging windows.

The development of GUIs should be *user centered*.

Goals: simple, aesthetic, productive, and customizable.

Characteristics of GUIs

The most significant features of a GUI are

Window: area of display device to view and interact with an object. Information is *viewed*.

Icon: pictorial representation of an object.

Menu: collection of choices, typically to perform actions on an object.

Pointer: is similar to a typing cursor, controlled by a pointing device, typically a mouse.

Client Area: area within a window where the user can enter data, strings or numbers.

Directness: direct manipulation of objects via the pointer, such as moving and dragging windows.

The development of GUIs should be *user centered*.

Goals: simple, aesthetic, productive, and customizable.

Characteristics of GUIs

The most significant features of a GUI are

Window: area of display device to view and interact with an object. Information is *viewed*.

Icon: pictorial representation of an object.

Menu: collection of choices, typically to perform actions on an object.

Pointer: is similar to a typing cursor, controlled by a pointing device, typically a mouse.

Client Area: area within a window where the user can enter data, strings or numbers.

Directness: direct manipulation of objects via the pointer, such as moving and dragging windows.

The development of GUIs should be *user centered*.

Goals: simple, aesthetic, productive, and customizable.

Graphical User Interfaces

characteristics and components

The Tk GUI Toolkit in Python

Tkinter, Tk, and Tcl
hello world again
using grid, Radiobutton,
and Checkbutton

Summary + Assignments

Characteristics of GUIs

The most significant features of a GUI are

Window: area of display device to view and interact with an object. Information is *viewed*.

Icon: pictorial representation of an object.

Menu: collection of choices, typically to perform actions on an object.

Pointer: is similar to a typing cursor, controlled by a pointing device, typically a mouse.

Client Area: area within a window where the user can enter data, strings or numbers.

Directness: direct manipulation of objects via the pointer, such as moving and dragging windows.

The development of GUIs should be *user centered*.

Goals: simple, aesthetic, productive, and customizable.

Characteristics of GUIs

The most significant features of a GUI are

Window: area of display device to view and interact with an object. Information is *viewed*.

Icon: pictorial representation of an object.

Menu: collection of choices, typically to perform actions on an object.

Pointer: is similar to a typing cursor, controlled by a pointing device, typically a mouse.

Client Area: area within a window where the user can enter data, strings or numbers.

Directness: direct manipulation of objects via the pointer, such as moving and dragging windows.

The development of GUIs should be *user centered*.

Goals: simple, aesthetic, productive, and customizable.

Characteristics of GUIs

The most significant features of a GUI are

Window: area of display device to view and interact with an object. Information is *viewed*.

Icon: pictorial representation of an object.

Menu: collection of choices, typically to perform actions on an object.

Pointer: is similar to a typing cursor, controlled by a pointing device, typically a mouse.

Client Area: area within a window where the user can enter data, strings or numbers.

Directness: direct manipulation of objects via the pointer, such as moving and dragging windows.

The development of GUIs should be *user centered*.

Goals: simple, aesthetic, productive, and customizable.

GUI components

GUI components are *widgets* = window gadgets.

Some GUI components are

Button: to trigger an event

Label: to display text or icons

Entry: to accept single line user input

Text: to accept multiple line user input

Menu: to display list of items

Listbox: to display list of text options

GUI components are *widgets* = window gadgets.
Some GUI components are

Button: to trigger an event

Label: to display text or icons

Entry: to accept single line user input

Text: to accept multiple line user input

Menu: to display list of items

Listbox: to display list of text options

GUI components are *widgets* = window gadgets.
Some GUI components are

Button: to trigger an event

Label: to display text or icons

Entry: to accept single line user input

Text: to accept multiple line user input

Menu: to display list of items

Listbox: to display list of text options

GUI components are *widgets* = window gadgets.
Some GUI components are

Button: to trigger an event

Label: to display text or icons

Entry: to accept single line user input

Text: to accept multiple line user input

Menu: to display list of items

Listbox: to display list of text options

GUI components are *widgets* = window gadgets.
Some GUI components are

Button: to trigger an event

Label: to display text or icons

Entry: to accept single line user input

Text: to accept multiple line user input

Menu: to display list of items

Listbox: to display list of text options

GUI components are *widgets* = window gadgets.
Some GUI components are

Button: to trigger an event

Label: to display text or icons

Entry: to accept single line user input

Text: to accept multiple line user input

Menu: to display list of items

Listbox: to display list of text options

GUI components are *widgets* = window gadgets.
Some GUI components are

Button: to trigger an event

Label: to display text or icons

Entry: to accept single line user input

Text: to accept multiple line user input

Menu: to display list of items

Listbox: to display list of text options

Outline

MCS 260 L-30

5 November 2007

Graphical User Interfaces

characteristics and components

Graphical User Interfaces

characteristics and components

The Tk GUI Toolkit in Python

Tkinter, Tk, and Tcl

hello world again

using grid, Radiobutton, and Checkbutton

The Tk GUI Toolkit in Python

Tkinter, Tk, and Tcl

hello world again
using grid, Radiobutton,
and Checkbutton

Summary + Assignments

Summary + Assignments

Tkinter, Tk, and Tcl

GUIs for Python programmers



The `Tkinter` (= Tk interface) library provides an object-oriented interface to the Tk GUI toolkit, the graphical interface development tool for `Tcl`, Tk = Tool Kit, `Tcl` = Tool Command Language.

Benefit: platform independent GUI development.

Graphical User Interfaces

characteristics and components

The Tk GUI Toolkit in Python

Tkinter, Tk, and Tcl

hello world again
using grid, Radiobutton,
and Checkbutton

Summary + Assignments

Tkinter, Tk, and Tcl

GUIs for Python programmers



The `Tkinter` (= Tk interface) library provides an object-oriented interface to the Tk GUI toolkit, the graphical interface development tool for `Tcl`, Tk = Tool Kit, `Tcl` = Tool Command Language.

Benefit: platform independent GUI development.

Graphical User Interfaces

characteristics and components

The Tk GUI Toolkit in Python

Tkinter, Tk, and Tcl

hello world again
using grid, Radiobutton,
and Checkbutton

Summary + Assignments

Graphical User Interfaces

characteristics and components

Graphical User Interfaces

characteristics and components

The Tk GUI Toolkit in Python

Tkinter, Tk, and Tcl

hello world again

using grid, Radiobutton, and Checkbutton

The Tk GUI Toolkit in Python

Tkinter, Tk, and Tcl

hello world again

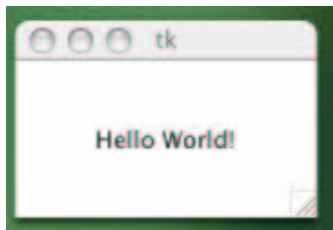
using grid, Radiobutton, and Checkbutton

Summary + Assignments

Summary + Assignments

Hello World!

our first GUI



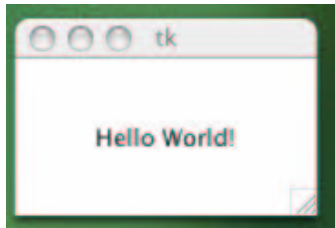
Five steps:

1. `from Tkinter import *`
2. create a new object of the class Tk
3. a label defines the text message
4. apply the geometry manager to the label
5. enter the main event loop

Until we close the window, the program stays in the loop.

Hello World!

our first GUI



Five steps:

1. `from Tkinter import *`
2. create a new object of the class Tk
3. a label defines the text message
4. apply the geometry manager to the label
5. enter the main event loop

Until we close the window, the program stays in the loop.

Graphical User Interfaces

characteristics and components

The Tk GUI Toolkit in Python

Tkinter, Tk, and Tcl

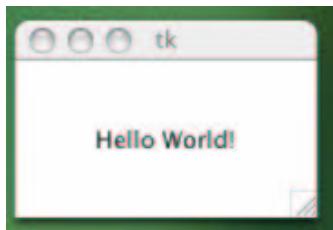
hello world again

using grid, Radiobutton, and Checkbutton

Summary + Assignments

Hello World!

our first GUI



Five steps:

1. `from Tkinter import *`
2. create a new object of the class Tk
3. a label defines the text message
4. apply the geometry manager to the label
5. enter the main event loop

Until we close the window, the program stays in the loop.

The Code for Hello World!

in the file `guihello.py`

```
# L-30 MCS 260 Mon 5 Nov 2007 : guihello.py
#
# Hello world with a Graphical User Interface.
# The code below displays "Hello World!" in a
# new window, using the Tkinter GUI library.
```

```
from Tkinter import *
top = Tk()          # top is the new window
# Label is a widget to design the interface
Lbl = Label(top,text="Hello World!",\
            width=20,height=5)
# to arrange the widget in a window we call
Lbl.pack()         # the geometry manager
top.mainloop()    # enter main event loop
```

Graphical User Interfaces

characteristics and components

The Tk GUI Toolkit in Python

Tkinter, Tk, and Tcl

hello world again

using grid, Radiobutton, and Checkbutton

Summary + Assignments

The Code for Hello World!

in the file `guihello.py`

```
# L-30 MCS 260 Mon 5 Nov 2007 : guihello.py
#
# Hello world with a Graphical User Interface.
# The code below displays "Hello World!" in a
# new window, using the Tkinter GUI library.
```

```
from Tkinter import *
top = Tk()          # top is the new window
# Label is a widget to design the interface
Lbl = Label(top,text="Hello World!",\
            width=20,height=5)
# to arrange the widget in a window we call
Lbl.pack()        # the geometry manager
top.mainloop()   # enter main event loop
```

The Code for Hello World!

in the file `guihello.py`

```
# L-30 MCS 260 Mon 5 Nov 2007 : guihello.py
#
# Hello world with a Graphical User Interface.
# The code below displays "Hello World!" in a
# new window, using the Tkinter GUI library.
```

```
from Tkinter import *
top = Tk()          # top is the new window
# Label is a widget to design the interface
Lbl = Label(top,text="Hello World!",\
            width=20,height=5)
# to arrange the widget in a window we call
Lbl.pack()        # the geometry manager
top.mainloop()   # enter main event loop
```

The Code for Hello World!

in the file `guihello.py`

```
# L-30 MCS 260 Mon 5 Nov 2007 : guihello.py
#
# Hello world with a Graphical User Interface.
# The code below displays "Hello World!" in a
# new window, using the Tkinter GUI library.
```

```
from Tkinter import *
top = Tk()          # top is the new window
# Label is a widget to design the interface
Lbl = Label(top,text="Hello World!",\
            width=20,height=5)
# to arrange the widget in a window we call
Lbl.pack()         # the geometry manager
top.mainloop()    # enter main event loop
```

Graphical User Interfaces

characteristics and components

The Tk GUI Toolkit in Python

Tkinter, Tk, and Tcl

hello world again

using grid, Radiobutton, and Checkbutton

Summary + Assignments

The Code for Hello World!

in the file `guihello.py`

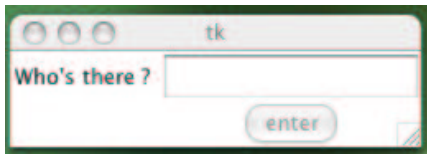
```
# L-30 MCS 260 Mon 5 Nov 2007 : guihello.py
#
# Hello world with a Graphical User Interface.
# The code below displays "Hello World!" in a
# new window, using the Tkinter GUI library.
```

```
from Tkinter import *
top = Tk()          # top is the new window
# Label is a widget to design the interface
Lbl = Label(top,text="Hello World!",\
            width=20,height=5)
# to arrange the widget in a window we call
Lbl.pack()        # the geometry manager
top.mainloop()    # enter main event loop
```

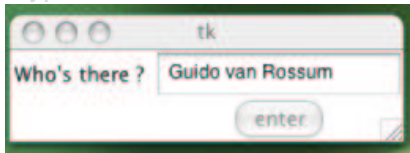
Dialogue with User

prompting for a name

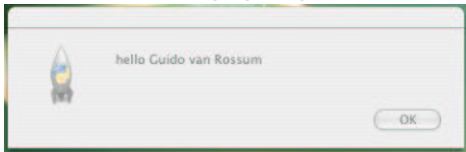
We prompt the user to enter a name:



After the user typed in a name:



Pushing the enter button pops up the window:



Graphical User
Interfaces

characteristics and
components

The Tk GUI Toolkit
in Python

Tkinter, Tk, and Tcl

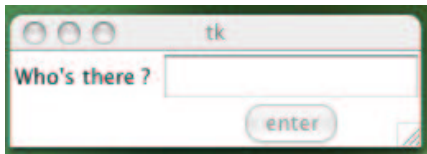
hello world again
using grid, Radiobutton,
and Checkbutton

Summary +
Assignments

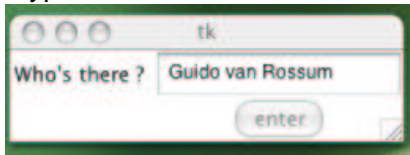
Dialogue with User

prompting for a name

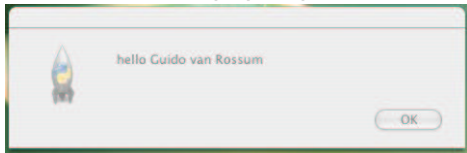
We prompt the user to enter a name:



After the user typed in a name:



Pushing the enter button pops up the window:



Graphical User
Interfaces

characteristics and
components

The Tk GUI Toolkit
in Python

Tkinter, Tk, and Tcl

hello world again

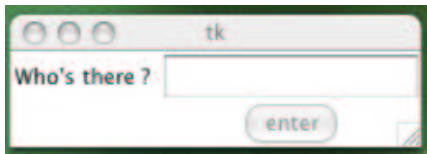
using grid, Radiobutton,
and Checkbutton

Summary +
Assignments

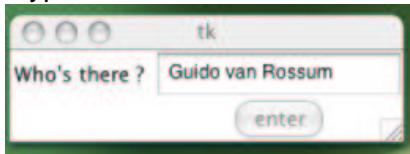
Dialogue with User

prompting for a name

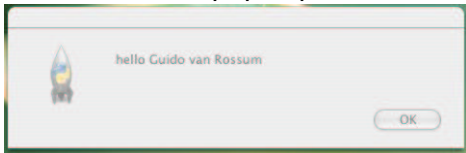
We prompt the user to enter a name:



After the user typed in a name:



Pushing the `enter` button pops up the window:



Graphical User
Interfaces

characteristics and
components

The Tk GUI Toolkit
in Python

Tkinter, Tk, and Tcl

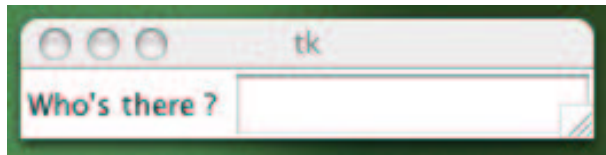
hello world again

using grid, Radiobutton,
and Checkbutton

Summary +
Assignments

The Widget Entry

to accept user input



```
from Tkinter import *
top = Tk()
Label(top, text="Who's there ? ").grid(row=0)
e = Entry(top)
e.grid(row=0, column=1)
top.mainloop()
```

Observe:

1. use of `row` and `column` with `grid`
2. the widget `Entry` to accept user strings

Graphical User Interfaces

characteristics and components

The Tk GUI Toolkit in Python

Tkinter, Tk, and Tcl

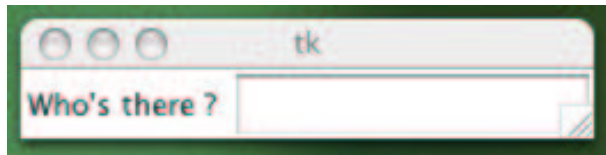
hello world again

using `grid`, `Radiobutton`, and `Checkbutton`

Summary + Assignments

The Widget Entry

to accept user input



```
from Tkinter import *
top = Tk()
Label(top, text="Who's there ? ").grid(row=0)
e = Entry(top)
e.grid(row=0, column=1)
top.mainloop()
```

Observe:

1. use of `row` and `column` with `grid`
2. the widget `Entry` to accept user strings

Graphical User Interfaces

characteristics and components

The Tk GUI Toolkit in Python

Tkinter, Tk, and Tcl

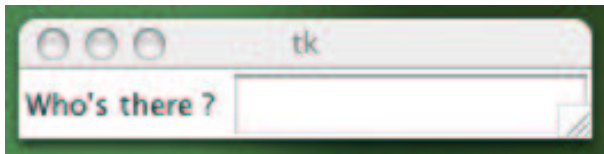
hello world again

using `grid`, `Radiobutton`, and `Checkbutton`

Summary + Assignments

The Widget Entry

to accept user input



```
from Tkinter import *
top = Tk()
Label(top, text="Who's there ? ").grid(row=0)
e = Entry(top)
e.grid(row=0, column=1)
top.mainloop()
```

Observe:

1. use of `row` and `column` with `grid`
2. the widget `Entry` to accept user strings

Graphical User Interfaces

characteristics and components

The Tk GUI Toolkit in Python

Tkinter, Tk, and Tcl

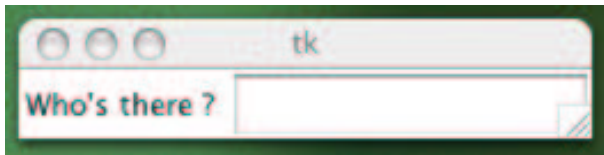
hello world again

using `grid`, `Radiobutton`, and `Checkbutton`

Summary + Assignments

The Widget Entry

to accept user input



```
from Tkinter import *
top = Tk()
Label(top, text="Who's there ? ").grid(row=0)
e = Entry(top)
e.grid(row=0, column=1)
top.mainloop()
```

Observe:

1. use of `row` and `column` with `grid`
2. the widget `Entry` to accept user strings

Graphical User Interfaces

characteristics and components

The Tk GUI Toolkit in Python

Tkinter, Tk, and Tcl

hello world again

using `grid`, `Radiobutton`, and `Checkbutton`

Summary + Assignments

The Widget Button

to enter user input



```
import tkinter as tk
def hello():
    "opens a window to say hello"
    s = 'hello ' + e.get()
    tkMessageBox.showinfo("enter",s)
b = Button(top,text="enter",command=hello)
b.grid(row=1,column=1)
top.mainloop()
```

1. Button will call hello when pressed
2. hello calls showinfo of tkinter

Graphical User Interfaces

characteristics and components

The Tk GUI Toolkit in Python

Tkinter, Tk, and Tcl

hello world again
using grid, Radiobutton,
and Checkbutton

Summary + Assignments

The Widget Button

to enter user input



```
import tkMessageBox
def hello():
    "opens a window to say hello"
    s = 'hello ' + e.get()
    tkMessageBox.showinfo("enter",s)
b = Button(top,text="enter",command=hello)
b.grid(row=1,column=1)
top.mainloop()
```

1. Button will call hello when pressed
2. hello calls showinfo of tkMessageBox

Graphical User Interfaces

characteristics and components

The Tk GUI Toolkit in Python

Tkinter, Tk, and Tcl

hello world again

using grid, Radiobutton, and Checkbutton

Summary + Assignments

The Widget Button

to enter user input



```
import tkMessageBox
def hello():
    "opens a window to say hello"
    s = 'hello ' + e.get()
    tkMessageBox.showinfo("enter",s)
b = Button(top,text="enter",command=hello)
b.grid(row=1,column=1)
top.mainloop()
```

1. Button will call hello when pressed
2. hello calls showinfo of tkMessageBox

Graphical User Interfaces

characteristics and components

The Tk GUI Toolkit in Python

Tkinter, Tk, and Tcl

hello world again

using grid, Radiobutton, and Checkbutton

Summary + Assignments

The Widget Button

to enter user input



```
import tkMessageBox
def hello():
    "opens a window to say hello"
    s = 'hello ' + e.get()
    tkMessageBox.showinfo("enter",s)
b = Button(top,text="enter",command=hello)
b.grid(row=1,column=1)
top.mainloop()
```

1. Button will call hello when pressed
2. hello calls showinfo of tkMessageBox

Graphical User
Interfaces

characteristics and
components

The Tk GUI Toolkit
in Python

Tkinter, Tk, and Tcl

hello world again

using grid, Radiobutton,
and Checkbutton

Summary +
Assignments

The Widget Button

to enter user input



```
import tkMessageBox
def hello():
    "opens a window to say hello"
    s = 'hello ' + e.get()
    tkMessageBox.showinfo("enter",s)
b = Button(top,text="enter",command=hello)
b.grid(row=1,column=1)
top.mainloop()
```

1. Button will call hello when pressed
2. hello calls showinfo of tkMessageBox

Graphical User
Interfaces

characteristics and
components

The Tk GUI Toolkit
in Python

Tkinter, Tk, and Tcl

hello world again

using grid, Radiobutton,
and Checkbutton

Summary +
Assignments

The Widget Button

to enter user input



```
import tkMessageBox
def hello():
    "opens a window to say hello"
    s = 'hello ' + e.get()
    tkMessageBox.showinfo("enter",s)
b = Button(top,text="enter",command=hello)
b.grid(row=1,column=1)
top.mainloop()
```

1. Button will call hello when pressed
2. hello calls showinfo of tkMessageBox

Graphical User
Interfaces

characteristics and
components

The Tk GUI Toolkit
in Python

Tkinter, Tk, and Tcl

hello world again

using grid, Radiobutton,
and Checkbutton

Summary +
Assignments

The Widget Entry and Button

in the file `guihello2.py`

```
from Tkinter import *
import tkMessageBox

top = Tk()
Label(top, text="Who's there ? ").grid(row=0)
e = Entry(top)
e.grid(row=0, column=1)

def hello():
    "opens a window to say hello"
    s = 'hello ' + e.get()
    tkMessageBox.showinfo("enter", s)

b = Button(top, text="enter", command=hello)
b.grid(row=1, column=1)
top.mainloop()
```

Graphical User
Interfaces

characteristics and
components

The Tk GUI Toolkit
in Python

Tkinter, Tk, and Tcl

hello world again

using grid, Radiobutton,
and Checkbutton

Summary +
Assignments

The Widget Entry and Button

in the file `guihello2.py`

```
from Tkinter import *
import tkMessageBox

top = Tk()
Label(top, text="Who's there ? ").grid(row=0)
e = Entry(top)
e.grid(row=0, column=1)

def hello():
    "opens a window to say hello"
    s = 'hello ' + e.get()
    tkMessageBox.showinfo("enter", s)

b = Button(top, text="enter", command=hello)
b.grid(row=1, column=1)
top.mainloop()
```

Graphical User
Interfaces

characteristics and
components

The Tk GUI Toolkit
in Python

Tkinter, Tk, and Tcl

hello world again

using grid, Radiobutton,
and Checkbutton

Summary +
Assignments

The Widget Entry and Button

in the file `guihello2.py`

```
from Tkinter import *
import tkMessageBox

top = Tk()
Label(top, text="Who's there ? ").grid(row=0)
e = Entry(top)
e.grid(row=0, column=1)

def hello():
    "opens a window to say hello"
    s = 'hello ' + e.get()
    tkMessageBox.showinfo("enter", s)

b = Button(top, text="enter", command=hello)
b.grid(row=1, column=1)
top.mainloop()
```

Graphical User
Interfaces

characteristics and
components

The Tk GUI Toolkit
in Python

Tkinter, Tk, and Tcl

hello world again

using grid, Radiobutton,
and Checkbutton

Summary +
Assignments

The Widget Entry and Button

in the file `guihello2.py`

```
from Tkinter import *
import tkMessageBox

top = Tk()
Label(top, text="Who's there ? ").grid(row=0)
e = Entry(top)
e.grid(row=0, column=1)

def hello():
    "opens a window to say hello"
    s = 'hello ' + e.get()
    tkMessageBox.showinfo("enter", s)

b = Button(top, text="enter", command=hello)
b.grid(row=1, column=1)
top.mainloop()
```

Outline

MCS 260 L-30

5 November 2007

Graphical User Interfaces

characteristics and components

Graphical User Interfaces

characteristics and components

The Tk GUI Toolkit in Python

Tkinter, Tk, and Tcl

hello world again

using grid, Radiobutton, and Checkbutton

The Tk GUI Toolkit in Python

Tkinter, Tk, and Tcl

hello world again

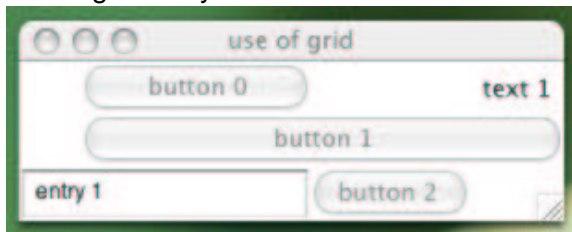
using grid, Radiobutton, and Checkbutton

Summary + Assignments

Summary + Assignments

the Geometry Manager grid

To layout the geometry like this



Observe:

1. the title of the window
2. buttons spanning multiple columns
3. an entry field containing text

Graphical User
Interfacescharacteristics and
componentsThe Tk GUI Toolkit
in PythonTkinter, Tk, and Tcl
hello world again
using grid, Radiobutton,
and CheckbuttonSummary +
Assignments

Using grid

the file usegrid.py

```
from Tkinter import *
top = Tk()
top.title("use of grid")
t1 = Label(top, text=" text 1 ")
t1.grid(row=0, column=4)
b0 = Button(top, text=" button 0 ")
b0.grid(row=0, column=1, sticky=W+E+N+S)
b1 = Button(top, text=" button 1 ")
b1.grid(row=1, column=1, columnspan=4, sticky=W+E+N+S)
e1 = Entry(top)
e1.insert(INSERT, "entry 1 ")
e1.grid(row=2, column=0, columnspan=2)
b2 = Button(top, text = "button 2 ")
b2.grid(row=2, column=3)
top.mainloop()
```

Graphical User
Interfacescharacteristics and
componentsThe Tk GUI Toolkit
in PythonTkinter, Tk, and Tcl
hello world again
using grid, Radiobutton,
and CheckbuttonSummary +
Assignments

Using grid

the file usegrid.py

```
from Tkinter import *
top = Tk()
top.title("use of grid")
t1 = Label(top, text=" text 1 ")
t1.grid(row=0, column=4)

b0 = Button(top, text=" button 0 ")
b0.grid(row=0, column=1, sticky=W+E+N+S)
b1 = Button(top, text=" button 1 ")
b1.grid(row=1, column=1, columnspan=4, sticky=W+E+N+S)
e1 = Entry(top)
e1.insert(INSERT, "entry 1 ")
e1.grid(row=2, column=0, columnspan=2)
b2 = Button(top, text = "button 2 ")
b2.grid(row=2, column=3)
top.mainloop()
```

Using grid

the file usegrid.py

```
from Tkinter import *
top = Tk()
top.title("use of grid")
t1 = Label(top,text=" text 1 ")
t1.grid(row=0,column=4)
b0 = Button(top,text=" button 0 ")
b0.grid(row=0,column=1,sticky=W+E+N+S)
b1 = Button(top,text=" button 1 ")
b1.grid(row=1,column=1,columnspan=4,sticky=W+E+N+S)
e1 = Entry(top)
e1.insert(INSERT,"entry 1 ")
e1.grid(row=2,column=0,columnspan=2)
b2 = Button(top,text = "button 2 ")
b2.grid(row=2,column=3)
top.mainloop()
```

Graphical User Interfaces

characteristics and components

The Tk GUI Toolkit in Python

Tkinter, Tk, and Tcl
hello world again
using grid, Radiobutton,
and Checkbutton

Summary + Assignments

Graphical User
Interfacescharacteristics and
componentsThe Tk GUI Toolkit
in PythonTkinter, Tk, and Tcl
hello world again
using grid, Radiobutton,
and CheckbuttonSummary +
Assignments

Using grid

the file usegrid.py

```
from Tkinter import *
top = Tk()
top.title("use of grid")
t1 = Label(top, text=" text 1 ")
t1.grid(row=0, column=4)
b0 = Button(top, text=" button 0 ")
b0.grid(row=0, column=1, sticky=W+E+N+S)
b1 = Button(top, text=" button 1 ")
b1.grid(row=1, column=1, columnspan=4, sticky=W+E+N+S)
e1 = Entry(top)
e1.insert(INSERT, "entry 1 ")
e1.grid(row=2, column=0, columnspan=2)
b2 = Button(top, text = "button 2 ")
b2.grid(row=2, column=3)
top.mainloop()
```

Using grid

the file usegrid.py

```
from Tkinter import *
top = Tk()
top.title("use of grid")
t1 = Label(top,text=" text 1 ")
t1.grid(row=0,column=4)
b0 = Button(top,text=" button 0 ")
b0.grid(row=0,column=1,sticky=W+E+N+S)
b1 = Button(top,text=" button 1 ")
b1.grid(row=1,column=1,columnspan=4,sticky=W+E+N+S)
e1 = Entry(top)
e1.insert(INSERT,"entry 1 ")
e1.grid(row=2,column=0,columnspan=2)
b2 = Button(top,text = "button 2 ")
b2.grid(row=2,column=3)
top.mainloop()
```

Graphical User Interfaces

characteristics and components

The Tk GUI Toolkit in Python

Tkinter, Tk, and Tcl
hello world again
using grid, Radiobutton,
and Checkbutton

Summary + Assignments

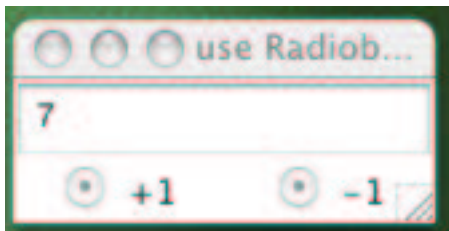
Graphical User
Interfacescharacteristics and
componentsThe Tk GUI Toolkit
in PythonTkinter, Tk, and Tcl
hello world again
using grid, Radiobutton,
and CheckbuttonSummary +
Assignments

Using grid

the file usegrid.py

```
from Tkinter import *
top = Tk()
top.title("use of grid")
t1 = Label(top, text=" text 1 ")
t1.grid(row=0, column=4)
b0 = Button(top, text=" button 0 ")
b0.grid(row=0, column=1, sticky=W+E+N+S)
b1 = Button(top, text=" button 1 ")
b1.grid(row=1, column=1, columnspan=4, sticky=W+E+N+S)
e1 = Entry(top)
e1.insert(INSERT, "entry 1 ")
e1.grid(row=2, column=0, columnspan=2)
b2 = Button(top, text = "button 2 ")
b2.grid(row=2, column=3)
top.mainloop()
```


Use of Radiobuttons



The GUI should do the following:

1. the entry field starts at 0
2. when user clicks on +1: add one
3. when user clicks on -1: subtract one

Code using Radiobuttons

in the file `guiradio.py`

```
from Tkinter import *
top = Tk()
top.title("use Radiobutton")
text = Entry(top)
text.insert(INSERT,"0") # initialization
text.grid(row=0,columnspan=2)

def plus():
    "Callback function, does +1"
    s = text.get() # data in Entry
    s = str(int(s) + 1) # add one to it
    text.delete(0,END) # clear Entry
    text.insert(INSERT,s) # insert result

add = Radiobutton(top,text="+1",\
    command=plus)
add.grid(row=1,column=0)
top.mainloop()
```

Code for the other Radiobutton "-1" is similar.

Graphical User Interfaces

characteristics and components

The Tk GUI Toolkit in Python

Tkinter, Tk, and Tcl

hello world again

using grid, Radiobutton, and Checkbutton

Summary + Assignments

Code using Radiobuttons

in the file `guiradio.py`

```

from Tkinter import *
top = Tk()
top.title("use Radiobutton")
text = Entry(top)
text.insert(INSERT,"0") # initialization
text.grid(row=0,columnspan=2)

def plus():
    "Callback function, does +1"
    s = text.get()          # data in Entry
    s = str(int(s) + 1)    # add one to it
    text.delete(0,END)    # clear Entry
    text.insert(INSERT,s) # insert result

add = Radiobutton(top,text="+1",\
    command=plus)
add.grid(row=1,column=0)
top.mainloop()

```

Code for the other Radiobutton "-1" is similar.

Graphical User Interfaces

characteristics and components

The Tk GUI Toolkit in Python

Tkinter, Tk, and Tcl
hello world again
using grid, Radiobutton,
and Checkbutton

Summary + Assignments

Code using Radiobuttons

in the file `guiradio.py`

```
from Tkinter import *
top = Tk()
top.title("use Radiobutton")
text = Entry(top)
text.insert(INSERT,"0") # initialization
text.grid(row=0,columnspan=2)

def plus():
    "Callback function, does +1"
    s = text.get()          # data in Entry
    s = str(int(s) + 1)    # add one to it
    text.delete(0,END)    # clear Entry
    text.insert(INSERT,s) # insert result

add = Radiobutton(top,text="+1",\
    command=plus)
add.grid(row=1,column=0)
top.mainloop()
```

Code for the other Radiobutton "-1" is similar.

Graphical User
Interfaces

characteristics and
components

The Tk GUI Toolkit
in Python

Tkinter, Tk, and Tcl
hello world again
using grid, Radiobutton,
and Checkbutton

Summary +
Assignments

Code using Radiobuttons

in the file `guiradio.py`

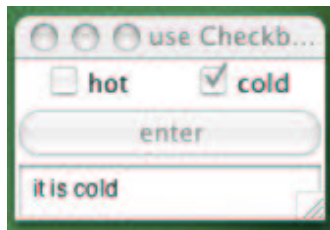
```
from Tkinter import *
top = Tk()
top.title("use Radiobutton")
text = Entry(top)
text.insert(INSERT,"0") # initialization
text.grid(row=0,columnspan=2)

def plus():
    "Callback function, does +1"
    s = text.get()          # data in Entry
    s = str(int(s) + 1)    # add one to it
    text.delete(0,END)    # clear Entry
    text.insert(INSERT,s) # insert result

add = Radiobutton(top,text="+1",\
    command=plus)
add.grid(row=1,column=0)
top.mainloop()
```

Code for the other Radiobutton "-1" is similar.

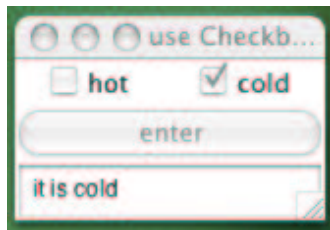
Using CheckButtons



Functionality:

1. the user can check one or two boxes
2. click on the enter button
3. to see a message displayed in the Entry box

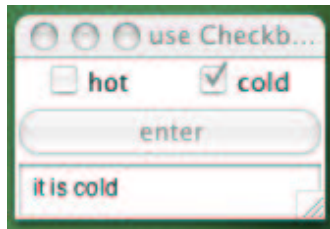
Using CheckButtons



Functionality:

1. the user can check one or two boxes
2. click on the enter button
3. to see a message displayed in the Entry box

Using CheckButtons



Functionality:

1. the user can check one or two boxes
2. click on the enter button
3. to see a message displayed in the Entry box

Code use Checkbutton

in the file guicheck.py

```
from Tkinter import *
top = Tk()
top.title("use Checkbutton")
H = IntVar() # determined by hot button
C = IntVar() # determined by cold button

hot = Checkbutton(top, text="hot", \
    variable = H, onvalue = 1, offvalue = 0)
hot.grid(row=0, column=0)

cold = Checkbutton(top, text="cold", \
    variable = C, onvalue = 1, offvalue = 0)
cold.grid(row=0, column=1)
```

H and C are variables, toggled on or off by the user selecting the corresponding box.

Graphical User Interfaces

characteristics and components

The Tk GUI Toolkit in Python

Tkinter, Tk, and Tcl

hello world again

using grid, Radiobutton, and Checkbutton

Summary + Assignments

Code use Checkbutton

in the file guicheck.py

```
from Tkinter import *
top = Tk()
top.title("use Checkbutton")
H = IntVar() # determined by hot button
C = IntVar() # determined by cold button

hot = Checkbutton(top, text="hot", \
    variable = H, onvalue = 1, offvalue = 0)
hot.grid(row=0, column=0)

cold = Checkbutton(top, text="cold", \
    variable = C, onvalue = 1, offvalue = 0)
cold.grid(row=0, column=1)
```

H and C are variables, toggled on or off by the user selecting the corresponding box.

Graphical User Interfaces

characteristics and components

The Tk GUI Toolkit in Python

Tkinter, Tk, and Tcl

hello world again

using grid, Radiobutton, and Checkbutton

Summary + Assignments

Code use Checkbutton

in the file guicheck.py

```
from Tkinter import *
top = Tk()
top.title("use Checkbutton")
H = IntVar() # determined by hot button
C = IntVar() # determined by cold button

hot = Checkbutton(top, text="hot", \
    variable = H, onvalue = 1, offvalue = 0)
hot.grid(row=0, column=0)

cold = Checkbutton(top, text="cold", \
    variable = C, onvalue = 1, offvalue = 0)
cold.grid(row=0, column=1)
```

H and C are variables, toggled on or off by the user selecting the corresponding box.

Graphical User
Interfacescharacteristics and
componentsThe Tk GUI Toolkit
in PythonTkinter, Tk, and Tcl
hello world again
using grid, Radiobutton,
and CheckbuttonSummary +
Assignments

code continued...

```
e = Entry(top)
e.grid(row=2,columnspan=2)

def act():
    "callback function for enter button"
    e.delete(0,END)
    if H.get() == 1 and C.get() == 0:
        e.insert(INSERT,"it is hot")
    if H.get() == 0 and C.get() == 1:
        e.insert(INSERT,"it is cold")
    if H.get() == 1 and C.get() == 1:
        e.insert(INSERT,"it is hot and cold")

b = Button(top,text = "enter",command=act)
b.grid(row=1,columnspan=2,sticky=W+E+N+S)
top.mainloop()
```

Observe the use of `H.get()` and `C.get()`.

code continued...

```
e = Entry(top)
e.grid(row=2,columnspan=2)

def act():
    "callback function for enter button"
    e.delete(0,END)
    if H.get() == 1 and C.get() == 0:
        e.insert(INSERT,"it is hot")
    if H.get() == 0 and C.get() == 1:
        e.insert(INSERT,"it is cold")
    if H.get() == 1 and C.get() == 1:
        e.insert(INSERT,"it is hot and cold")

b = Button(top,text = "enter",command=act)
b.grid(row=1,columnspan=2,sticky=W+E+N+S)
top.mainloop()
```

Observe the use of `H.get()` and `C.get()`.

code continued...

```
e = Entry(top)
e.grid(row=2,columnspan=2)

def act():
    "callback function for enter button"
    e.delete(0,END)
    if H.get() == 1 and C.get() == 0:
        e.insert(INSERT,"it is hot")
    if H.get() == 0 and C.get() == 1:
        e.insert(INSERT,"it is cold")
    if H.get() == 1 and C.get() == 1:
        e.insert(INSERT,"it is hot and cold")

b = Button(top,text = "enter",command=act)
b.grid(row=1,columnspan=2,sticky=W+E+N+S)
top.mainloop()
```

Observe the use of `H.get()` and `C.get()`.

Graphical User
Interfaces

characteristics and
components

The Tk GUI Toolkit
in Python

Tkinter, Tk, and Tcl
hello world again
using grid, Radiobutton,
and Checkbutton

Summary +
Assignments

code continued...

```
e = Entry(top)
e.grid(row=2,columnspan=2)

def act():
    "callback function for enter button"
    e.delete(0,END)
    if H.get() == 1 and C.get() == 0:
        e.insert(INSERT,"it is hot")
    if H.get() == 0 and C.get() == 1:
        e.insert(INSERT,"it is cold")
    if H.get() == 1 and C.get() == 1:
        e.insert(INSERT,"it is hot and cold")

b = Button(top,text = "enter",command=act)
b.grid(row=1,columnspan=2,sticky=W+E+N+S)
top.mainloop()
```

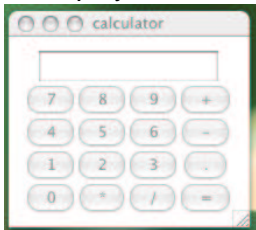
Observe the use of `H.get()` and `C.get()`.

Summary + Assignments

We started Chapter 15 in *Making Use of Python*.

Assignments:

1. Write Python code to display:



You should not provide any functionality.

2. Add functionality to the calculator shown above.
3. Design a GUI to convert temperatures between Fahrenheit and Celsius. Draw the layout and decide what widgets you will use.
4. Give Python code for the previous exercise.

Graphical User Interfaces

characteristics and components

The Tk GUI Toolkit in Python

Tkinter, Tk, and Tcl
hello world again
using grid, Radiobutton,
and Checkbutton

Summary + Assignments