# Week 2, Examples 2

```python
# 1.py

# GraphWin is a class. Think of it as code that you can use to create
# GraphWin objects

# What is a GraphWin object? It is a particular instantiation of the class.

# For example if there was a class (i.e., code) to create student objects,
# you can use this class code to instantiate students Bill, Bob, and Tom
# Now Bill, Bob and Tom would be student objects.

# The GraphWin object is a graphics window. You acquire it from the
# graphics.py library, just like you got math.sqrt from math.py

# The GraphWin class has variables, and it uses those variables to give
# you different kinds of windows.

def wait():
    dummyvar = input(" ");

from graphics import *  # instead of just import, in which case you call
# functions using "graphics.GraphWin", like
math.sqrt  # using this kind of import means no need to say
# "graphics." each time. Just GraphWin will do.

def main():
    wait()
    w1 = GraphWin("Small",200,200)  # (200,200) is default size
    wait()
    w2 = GraphWin("Medium",400,400)
    wait()
    w3 = GraphWin("Large",800,800)
    wait()
    w1.close()  # notice that w1,w2,w3 are objects and they each have
                # methods
    w2.close()  # that can work with them. These "methods" are functions
    you
    w3.close()  # call from the GraphWin class
```

V. Rego, Sept 3, 2015
wait()
for i in range(1,100,1):

    w = GraphWin("Oops! Yet another graphics window",10*i,10*i)

    # Note that w keeps getting reassigned to point to a new object as
    # the loop index changes
    # and note that we did not call the w.close() function to close windows

    # this was just to show you the windows
    # its bad programming practice to reassign w without closing the old w

    # w = GraphWin("Window 1",10,10)
    # now do a w.close()
    # or else when, without closing, you reassign variable w
    # w = GraphWin("Window 2",20,20)

    # you have lost the way to close the 10x10 window because w now
    # refers to the 20x20 window

    # so always be careful of your variables and losing information

#2.py

#Lets do the window example again, but close each window before
#opening a new one. So you will not see a "cascade" of windows.

from graphics import *  # instead of just import, in which case you call
                        # functions using "graphics.GraphWin", like
math.sqrt  # using this kind of import means no need to say
            # "graphics." each time. Just GraphWin will do.

def main():

    for i in range(1,100,1):

        w = GraphWin("Oops! Yet another graphics window",10*i,10*i)

        w.close()  # close the window before changing w in next loop iteration
# graphics.py gives you other classes besides Graphwin

# you can access classes to give you circles, lines, rectangle, polygons etc
# inside any graphics window

```python
def wait():
    dummy = input(" ")

from graphics import *  # get access to all of graphics.py's functions

import time

def main():

    wait()  # first get a window
    w = GraphWin("Example Window", 800, 800)  # (0,0) is at top left corner
    w.setCoords(0, 0, 799, 799)  # (799,799) is at bottom-right corner
    # x is horizontal, y is vertical
    # in (x,y)

    wait()  # draw a circle

    center = Point(100, 100)  # place a point at location (100,100)
    c = Circle(center, 90)
    c.draw(w)  # call method to make circle in window w

    wait()
    lec = Point(65, 60)  # left eye center

    le = Circle(lec, 20)
    le.draw(w)

    wait()
    rec = Point(135, 60)  # using symmetry about (100,100) center
    re = Circle(rec, 20)
    re.draw(w)

    wait()

    # put a horizontal line in middle of right eye

    droopy = Line(Point(115, 60), Point(155, 60))  # now the right eye looks
```

Computer Science Courses - http://courses.cs.purdue.edu/
droopy
droopy.draw(w)

# now you can turn that eye into an ice-cream cone :) by drawing horizontal lines
wait()

for i in range(0,96,1):
    droopy.setFill("black")
    droopy.setWidth(1)
    droopy = Line(Point(115+(i/10.0),60+i),Point(155-(i/10.0),60+i))
    droopy.draw(w)

wait()
#---------------------
# Now a rectangle

r = Rectangle(Point(250,250), Point(600,450))
r.draw(w)

# and some lines

wait()

vert = Line(Point(450,250),Point(450,450))
vert.draw(w)

wait()

horz = Line(Point(250,340),Point(600,340))
horz.draw(w)

wait()
#---------------------
# Now a triangle, via the polygon class, on the left hand side of page bottom

t1 = Polygon(Point(225,550),Point(25,750),Point(425,750))
t1.draw(w)

# if you say t2 = t1, you simply create a new name t2 for an object that is
# already called t1, and you can access the object via either name.

# BUT if you wanted to create a duplicate object instead, say another such
# triangle, then you must CLONE it using the "clone()" method

wait()
# How to copy any object, in this case the triangle

t2 = t1.clone()  # now a COPY of t1 is made, and t2 is not pointing to t1

# the copy is ready and sits atop the old object; we need to move t

# lets move it some distance to the right and a bit up

t2.move(350,-75)  # 350 units to right, and 75 units up (hence minus)
t2.draw(w)

#label each triangle, so we can tell from the pic

wait()

t1text = Text(Point(230,650),"t1 is the original")
t1text.setSize(18)
t1text.draw(w)

wait()

t2text = Text(Point(580,625),"t2 is the clone")
t2text.setSize(18)
t2text.draw(w)

#------------------

# You can draw and undraw things; let's try with a line

l = Line(Point(220,150),Point(450,10))
l.draw(w)

for i in range(1,600,+1):
    time.sleep(0.05)  # function from time library; it puts the program to sleep for a bit
    l.undraw()
    l = Line(Point(220+i,150+i),Point(450+i,10+i))  # draw new line, bit lower
    l.setFill("blue")
l.draw(w)

# Because we draw a line, let it stay drawn for 0.05 secs
# and then "undraw" (erase) it and draw the same line a little lower down

# your brain/eyes fool you into thinking the original line is moving :)

---

#4.py

# Prof. R had a VERY scary dream where he was attacked by a man with a
horrid
# red face and blue nose. The police heard about it and wanted a
description.
# He decided to use the graphics lib with objects and colors to accurately
# describe what he saw :)

from graphics import *  #import all the functions, no need to type x.func
now

def main():

    w = GraphWin("Gregory Peck",800,800)  # 800x800 box

    # Remember the top left corner is (0,0), i.e., x = 0, y = 0, and
    # the bottom right corner is (799,799), i.e., x = 799, y = 799.

    # As x ix increased you move to the right
    # As y is increased you move down

    cen = Point(400,400)        # this is the big red circle for face
cir = Circle(cen,350)
cir.setFill("red")
cir.draw(w)

    leye_cen = Point(250,350)   # yellow part of left eye
leye = Circle(leye_cen,45)
leye.setFill("yellow")
leye.draw(w)

    leyeball_cen = Point(250,350) # left eyeball (black)
leyeball = Circle(leye_cen,20)
leyeball.setFill("black")
leyeball.draw(w)

    reye_cen = Point(550,325)   # big black circle for right eye"
reye = Circle(reye_cen,100)
reye.setFill("black")
reye.draw(w)

    # The next rectangle covers the top of the right eye. By making it red,
    # it blends with face colour and creates a flat eyepatch top. Set line
    # width to 0 so that the rectangle border cannot be seen

    rect = Rectangle(Point(425,225), Point(675,295))
rect.setFill("red")  # set colour to "blue" to see this rectangle
rect.setWidth(0)     # don't want rectangle to show boundary
rect.draw(w)
#5.py

#IMPORTANT: instead of counting points from the top left corner in a
#graph window, python makes things easier by letting you define your
#own coordinate system, as large or as small as you want, inside the
#graphics window!

#Suppose a mutual-fund manager from Snooty & Co. wants to report his annual
#profit from his top tech investments to his clients:

# Here is the data
# $5.2M GOOG
# $3.1M YHOO
# $4.5M INTC
# $7.7M AAPL
# $3.8M QCOM
# $1.75 ADBE

# and we want to plot these sizes in a bar chart, in increasing order.

def wait():  # this is only used to pause in class
dummyvar = input(" ")

from graphics import *

def main():

    wait()
w = GraphWin("Annual Tech. Investment Profits",800,800)

    w.setCoords(0.0,0.0,14.0,14.0) # IMPORTANT: we redefine the coord. system!

    xmin = 2.0       # this is the (0,0)
ymin = 3.0       # corner of the graph inside the w window

    xmax = 12.0     # this is the right hand
    ymax = 14.0     # top corner of the graph inside the w window

    wait()

    horzline = Line(Point(xmin,ymin),Point(xmax,ymin))
    horzline.setWidth(3)
    horzline.draw(w)

    wait()

    # First the Heading under the chart

    tag = Text(Point(6.0,1),"Snooty and Co. Annual Tech Profits")
    tag.setStyle("bold")
    tag.setSize(20)
    tag.setTextColor("green")
    tag.draw(w)

    wait()

    #We have 14-2 = 12 units to work with on the x-axis to place tall bars
#skip the first 0.5 units, and then
# repeat this: place a tall bar 1 unit wide and skip 0.5 units
# 6 times

#------- ADBE
x = 0.5 + xmin
y = ymin
adbe = Rectangle(Point(x,ymin),Point(x+1,ymin + 1.75))
adbe.setFill("blue")
adbe.draw(w)
wait()
t1 = Text(Point(x+0.5,ymin-0.5),"ADBE")
t1.setSize(20)
t1.setStyle("bold")
t1.setTextColor("blue")
t1.draw(w)
wait()
tt1 = Text(Point(x+0.5,ymin+1.75+0.5),"$1.75M")
tt1.setSize(15)
tt1.setTextColor("blue")
tt1.draw(w)

#------- YHOO
x = x + 1.5

yhoo = Rectangle(Point(x,ymin),Point(x+1,ymin + 3.1))
yhoo.setFill("red")
yhoo.draw(w)
wait()
t2 = Text(Point(x+0.5,ymin-0.5),"YHOO")
t2.setSize(20)
t2.setStyle("bold")
t2.setTextColor("red")
t2.draw(w)
wait()
tt1 = Text(Point(x+0.5,ymin+3.10+0.5),"$3.10M")
tt1.setSize(15)
tt1.setTextColor("red")
tt1.draw(w)

# For homework, draw the other 4. Cut and paste from YHOO, and change
# the numbers and colors accordingly for each