class CarWashMachine(object):

""" interface to the CarWashMachine"""

def performWash(self):

"""signal machinery to wash"""

print "WASHING: scrub a dub"

def performTouchless(self):

"""signal machinery to touchless wash"""

print "TOUCHLESS: scrub a dub"

def performSoak(self):

"""signal machinery to soak"""

print "SOAKING: splish splash"

def performWax(self):

"""signal machinery to wax"""

print "WAXING : rub a dub"

def isWorking(self):

"""condition returned by equipment"""

return True

class FroBozzWash(CarWashMachine):

"""extends CarWashMachine, implements business logic"""

def simple(self):

"""Action Sent To Wash Machine: Wash"""

self.performWash()

print "\*\*\* Operations Complete \*\*\*"

def clean(self):

"""Actions Sent To Wash Machine: Wash, Soak"""

self.performWash()

self.performSoak()

print "\*\*\* Operations Complete \*\*\*"

def stupendous(self):

"""Actions Sent To Wash Machine: Wash, Soak, Wax"""

self.performWash()

self.performSoak()

self.performWax()

print "\*\*\* Operations Complete \*\*\*"

def tsimple(self):

"""Actions Sent To Wash Machine: Touchless"""

self.performTouchless()

print "\*\*\* Operations Complete \*\*\*"

def tclean(self):

"""Actions Sent To Wash Machine: Touchless, Soak"""

self.performTouchless()

self.performSoak()

print "\*\*\* Operations Complete \*\*\*"

def tstupendous(self):

"""Actions Sent To Wash Machine: Touchless, Soak, Wax"""

self.performTouchless()

self.performSoak()

self.performWax()

print "\*\*\* Operations Complete \*\*\*"

class CoinBox(object):

"""implements money taker/dispenser"""

def \_init\_(self):

"""initialize dollars at at startup"""

self.dollars = 0

self.coins = 0

def insert\_dollar(self):

"""a dollar is accepted by the money taker"""

self.dollars += 1

def insert\_coin(self):

"""a coint is accepted by the money taker"""

self.coins += 1

def return\_dollars(self):

"""return dollars(int) to user, re-init dollars for next user"""

dollars = self.dollars

self.dollars = 0

return dollars

def return\_coins(self):

"""return coins(int) to user, re-init coins for next user"""

coins = self.coins

self.coins = 0

return coins

def use\_dollars(self, dollars\_required):

"""if dollars >= dollars\_required, decrement dollars return true,

else return false"""

if self.dollars >= dollars\_required:

self.dollars -= dollars\_required

return True

else:

return False

def use\_coins(self, coins\_required):

"""if coins >= coins\_required, decrement coins return true,

else return false"""

if self.coins >= coins\_required:

self.coins -= coins\_required

return True

else:

return False

def usermenu(current\_dollars, current\_coins):

"""present user interface, return users requested Action"""

print "Your balance: $%i | %i coins" % (current\_dollars, current\_coins)

print

print "[i] insert dollar"

print "[c] insert coin"

print "[1] Buy Simple ($5/1 Coin)"

print "[2] Buy Clean ($6/2 Coins)"

print "[3] Buy Stupendous ($7/3 Coins)"

print "[4] Buy Touchless Simple ($6/1 Coin)"

print "[5] Buy Touchless Clean ($7/2 Coins)"

print "[6] Buy Touchless Stupendous ($8/3 Coins)"

print "[q] Quit"

return raw\_input('Action :')

#make some objects

FBW = FroBozzWash()

COINBOX = CoinBox()

#populate the PRICELIST with type/price/action-method information

PRICELIST = [

{'type':'Simple', 'price':5, 'coins':1, 'action':FBW.simple},

{'type':'Clean', 'price':6, 'coins':2, 'action':FBW.clean},

{'type':'Stupendous', 'price':7, 'coins':3, 'action':FBW.stupendous},

{'type':'Touchless Simple','price':6, 'coins':1, 'action':FBW.tsimple},

{'type':'Touchless Clean', 'price':7, 'coins':2, 'action':FBW.tclean},

{'type':'Touchless Stupendous','price':8, 'coins':3,

'action':FBW.tstupendous}

]

ACTION = '' #put the needle on the record

while ACTION != 'q': #start dancing

ACTION = usermenu(COINBOX.dollars, COINBOX.coins)#defer to puny human

#respond to request

if ACTION == 'i': # dollar inserted

COINBOX.insert\_dollar() #chalk it up

print 'Cha-Ching' #output warm fuzzy

if ACTION == 'c': #coin inserted

COINBOX.insert\_coin() #add it up

print 'Cha-Chang'

elif ACTION in ['1', '2', '3', '4', '5', '6']: #machine action selected

if FBW.isWorking(): # only attempt if machine is on-line

IDX = int(ACTION) - 1 # make an int index out of the ACTION

print "You Selected: %s" % PRICELIST[IDX]['type']

if COINBOX.use\_dollars(PRICELIST[IDX]['price']): #charge it!

PRICELIST[IDX]['action']() #call the method

elif COINBOX.use\_coins(PRICELIST[IDX]['coins']):

PRICELIST[IDX]['action']() #call the method

else: #they need to get J-O-B if they want to wash with me

print "Sorry, Insert More Money/Coins or Try different Action"

else: # machine if off line

print "Sorry, The Car Wash is temporarily off-line"

print "Please try again later."

elif ACTION == 'q': #they've quit

if COINBOX.dollars: # give'm back their dime

print "Returning $%i" % COINBOX.return\_dollars()

if COINBOX.coins:

print "Returning %i Coins" % COINBOX.return\_coins()

else: #let them know you are not equipped with telepathic module

print "Sorry, requested Action not understood"

print