```
#-----Statement of Authorship----------------------------------------#
#
#  This is an individual assessment item.  By submitting this
#  code I agree that it represents my own work.  I am aware of
#  the University rule that a student must not act in a manner
#  which constitutes academic dishonesty as stated and explained
#  in QUT's Manual of Policies and Procedures, Section C/5.3
#  "Academic Integrity" and Section E/2.1 "Student Code of Conduct".
#
#    Student no: n9983244
#    Student name: John Santias
#
#  NB: Files submitted without a completed copy of this statement
#  will not be marked.  Submitted files will be subjected to
#  software plagiarism analysis using the MoSS system
#  (http://theory.stanford.edu/~aiken/moss/).
#
#--------------------------------------------------------------------#



#-----Assignment Description-----------------------------------------#
#
#  Online Shopper
#
#  In this assignment you will combine your knowledge of HTMl/XML
#  mark-up languages with your skills in Python scripting, pattern
#  matching, and Graphical User Interface design to produce a useful
#  application for aggregating product data published by a variety of
#  online shops.  See the instruction sheet accompanying this file
#  for full details.
#
#--------------------------------------------------------------------#



#-----Imported Functions---------------------------------------------#
#
# Below are various import statements for helpful functions.  You
# should be able to complete this assignment using these
# functions only.  Note that not all of these functions are
# needed to successfully complete this assignment.

# The function for opening a web document given its URL.
# (You WILL need to use this function in your solution.)
from urllib import urlopen

# Import the standard Tkinter functions. (You WILL need to use
# these functions in your solution.)
from Tkinter import *

# Functions for finding all occurrences of a pattern
# defined via a regular expression.  (You do NOT need to
# use these functions in your solution, although you will find
# it difficult to produce a robust solution without using
# regular expressions.)
from re import findall, finditer
```

```
# Import the standard SQLite functions just in case they're
# needed.
from sqlite3 import *

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



#-----Student's Solution---------------------------------------------#
#
# Put your solution at the end of this file.
#

#URLs
#Category 1 = DVDs
url1 = 'http://www.fishpond.com.au/Movies'
#Category 2 = Games
url2 = 'http://www.gamesparadise.com.au/games-sale'
#Category 3 = Music
url3 = 'http://www.ellaways.com.au/guitars-amps-effects.html'

#Read the contents of the web page as a character string
web_page1 = urlopen(url1)
web_page2 = urlopen(url2)
web_page3 = urlopen(url3)

#Extract the web page's content
html1 = web_page1.read()
html2 = web_page2.read()
html3 = web_page3.read()

#<-----------------------EXTRACT FROM URL 1----------------------->
Cost = []
Item_from_one = []
Purchased_Items = []
Price_Item = []
Total = 0
import random #To choose a random item

def site_one():
#Find and select an image from url1
      image1 = findall('<img.* src="http:\/\/rcdn-([^"]+)".*>', html1)

      links1 = []
      for link in image1:
                 links = "http://rcdn-" + link
                 links1.append(links)

#Choose a random link image
      chosen_link1 = []
      import random
      chosen_link1.append(random.choice(links1))

#Find the title that matches the image
      title1 = []
      title = findall('(?<=' + chosen_link1[0] + '\"\sborder=\"0\"
alt=\").*?(?=\")'\
            , html1)
      title1.append(title)

#Get the price matching the item
#Put backslash on title for regex
      search1 = title1[0][0]
      search1 = str.replace(search1, '[', '\[')
      search1 = str.replace(search1, ']', '\]')
#Get the price
      getting_price = []
      getting_price.append(findall('(' + search1 + '(.*\n){16})', html1))
      the_actual_matchin_cost1 =
findall('(?<="productSpecialPrice"><b>\$)[0-9]*.[0-9]*',\
                                             getting_price[0][0][0]) #For
items with <b> tag
      the_actual_matchin_cost2 =
findall('(?<="productSpecialPrice">\$)[0-9]*.[0-9]*',\
                                             getting_price[0][0][0]) #For
items without <b> tag

      cost = []

      if len(the_actual_matchin_cost1) >= 1:
            cost.append(the_actual_matchin_cost1[0])
      elif len(the_actual_matchin_cost2) >= 1:
            cost.append(the_actual_matchin_cost2[0])
      else:
            pass

#Combine the title, image and price
      Combined_item1 = []
      Combined_item1.append(title1[0])
      Combined_item1.append(chosen_link1[0])
      Combined_item1.append(cost[0])

#Flatten the list of list (Combined_item1)
      Item_from_one = [subitem for item in Combined_item1 for subitem in
\
                          (item if isinstance(item, list) else [item])]
      Purchased_Items.append(Item_from_one)
      Price_Item.append(Item_from_one)

#<------------------END OF EXTRACTION FROM URL 1------------------>
#<-----------------------EXTRACT FROM URL 2----------------------->

Item_from_two = []
def site_two():

#Get all items with their description, price and image
      item2 = []
      find_item2 = findall('<div class="product-image-wrapper"
style="max-width:295px;">((\n.*){30})',\
                              html2)
      item2.append(find_item2)

#Select an item
```

```
      selected2 = []
      selected2.append(random.choice(item2[0]))

#Get the title of the item
      title2 = []
      Game = findall('(?<=title=").*(?=\".class="product-image)',
selected2[0][0])
      title2.append(Game)

#Get the image of the item
      image2 = []
      find_img = findall('(?<="product-image">\\n.{24}<img src=").*(?="
alt=)',\
                           selected2[0][0])
      image2.append(find_img)

#Get the price of the item
      price2 = []
      find_price = findall('(?<=\$)[0-9]+\.[0-9]+', selected2[0][0])
      price2.append(find_price)

#Combine price title and image
      Combined_item2 = []
      Combined_item2.append(title2[0])
      Combined_item2.append(image2[0])
      Combined_item2.append(price2[0])

#Flatten the list of list (Combined_item2)
      Item_from_two = [subitem for item in Combined_item2 for subitem in\
                         (item if isinstance(item, list) else [item])]
      Purchased_Items.append(Item_from_two)
      Price_Item.append(Item_from_two)

#<-------------------END OF EXTRACTION FROM URL 2------------------>
#<------------------------EXTRACT FROM URL 3----------------------->

Item_from_three = []

def site_three():

#Find and select an image from url1
      item3 = []
      find_item3 = findall('(?<=245x245/).*?(?=\")', html3)
      item3.append(random.choice(find_item3))

#Add the two links together
      chosen_link3 =
"http://www.ellaways.com.au/media/catalog/product/cache/1/small_image/245
x245/"\
                       + item3[0]

# Find the title that matches with the image
      title3 = []
      link =
"http://www.ellaways.com.au/media/catalog/product/cache/1/small_image/245
x245/" \
                       #Link without the actual product number/code
      object3 = findall('(?<=' + link + item3[0] + '\"\salt=\").*(?=\")',
html3)
      title3.append(object3)

#Get the price of the item
      multilines = []
      Get_lines = findall('(' + chosen_link3 + '(.*\n){28})', html3)
      multilines.append(Get_lines)
      findcost1 = findall('(?<=class=\"price\">\$)[0-9]+\,\d*\.\d*',
multilines[0][0][0]) #finds price with commas
      findcost2 = findall('(?<=class=\"price\">\$)\d*\.\d*',
multilines[0][0][0]) #finds price without commas
      find_specialprice_RRP1 = findall('(?<=<span class=\"price\"
itemprop=\"price\".id=\"product-price-.{5}\">\\n.{20}\$)[0-9]+,[0-9]+.[0-
9]+',\
                                          multilines[0][0][0])#finds
special price with commas
      find_specialprice_RRP2 = findall('(?<=<span class=\"price\"
itemprop=\"price\".id=\"product-price-.{5}\">\\n.{20}\$)[0-9]+\.[0-9]+',\
                                          multilines[0][0][0])#finds
special price without comma

#Append price to cost3
      cost3 = []
      if len(findcost1) >= 1:
            cost3.append(findcost1[0])
      elif len(findcost2) >= 1:
            cost3.append(findcost2[0])
      elif len(find_specialprice_RRP1) >= 1:
            cost3.append(find_specialprice_RRP1[0])
      elif len(find_specialprice_RRP2) >= 1:
            cost3.append(find_specialprice_RRP2[0])
      else:
            pass

#Take off comma in price
      cost3 = str.replace(cost3[0], ',', '')

#Combine price title and image
      Combined_item3 = []
      Combined_item3 = title3
      Combined_item3.append(chosen_link3)
      Combined_item3.append(cost3)

#Flatten the list of list (Combined_item3)
      Item_from_three = [subitem for item in Combined_item3 for subitem
in\
                            (item if isinstance(item, list) else [item])]
      Purchased_Items.append(Item_from_three)
      Price_Item.append(Item_from_three)

#<-------------------END OF EXTRACTION FROM URL 3------------------>
#<------------------GENERATE THE HTML FILE INVOICE----------------->
message1_1 = """
<!DOCTYPE html>
<html>
<head>
      <title>Entertainment World Invoice</title>
```

```
<style>
table {
    border: 1px solid black;
}
</style>
</head>
      <body>
            <center><h1>Entertainment World Invoice</h1>
            <img
src="https://www.robi.com.bd/files/large/7ea7f457b8f884c" width="225"
height ="200">
            <h2>Thank you for shopping with us</h2>
            <h2>Total for the purchases below:</h2>
            <h2>$
"""

message1_2a = """
            <small>AUD</small></h2></center>
            <center><table border="3">
            <tr>
            """

message1_2 = ""

message1_3 = """
            </table></center>
            <center><h3>Please come again</h3>
            <p><i>Entertainment world</i> is sponsored by:</p></center>
            <div style="width: 25%; margin-left: auto; margin-right:
auto;">
                  <ul>
                        <li><a
href="http://www.fishpond.com.au/Movies"</a>http://www.fishpond.com.au/Mo
vies</li>
                        <li><a
href="http://www.gamesparadise.com.au/games-
sale"</a>http://www.gamesparadise.com.au/games-sale</li>
                        <li><a href="http://www.ellaways.com.au/guitars-
amps-effects.html"</a>http://www.ellaways.com.au/guitars-amps-
effects.html</li>
                  </ul>
            </div>
      </body>
</html>"""

message2 = """
<!DOCTYPE html>
<html>
<head>
      <title>Entertainment World Invoice</title>
</head>
      <body><center>
            <h1>Entertainment World Invoice</h1>
            <img
src="https://www.robi.com.bd/files/large/7ea7f457b8f884c" width="225"
height ="200">
            <h2>Thank you for browsing with us</h2>
            <h2>No Charge</h2>
            <center><h3>Please come again</h3>
            <p><i>Entertainment world</i> is sponsored by:</p></center>
            <div style="width: 25%; margin-left: auto; margin-right:
auto;">
                  <ul>
                        <li><a
href="http://www.fishpond.com.au/Movies"</a>http://www.fishpond.com.au/Mo
vies</li>
                        <li><a
href="http://www.gamesparadise.com.au/games-
saleg"</a>http://www.gamesparadise.com.au/games-sale</li>
                        <li><a href="http://www.ellaways.com.au/guitars-
amps-effects.html"</a>http://www.ellaways.com.au/guitars-amps-
effects.html</li>
                  </ul>
            </div>
      </body>
</html>"""

##<--------------------ADD ITEMS INTO HTML DOC-------------------->

def tables():
      result = ""
      item_count = len(Purchased_Items)
      row_count = 0

#Put the selected items into tables
      if len(Purchased_Items) > 1:
            all = []
            for i in Purchased_Items:
                  i[-1]
                  result = result + "<td width=\"225\"
height=\"300\">\n<center><strong>" +\
                                         i[0] + "</strong></center>\n"
                  result = result + "<br>\n<center><img src=\"" \
                                         + i[1] + "\" height=\"177\"
width=\"124\"></center>\n</br>\n"
                  result = result + "<br>\n<center>" + "Our Price: $" +\
                                         i[2] + "<small>AUD</small>"
"\n</br></center>\n</td>\n"
                  row_count = row_count + 1

#Make new row when there are two items in a row
                  if row_count % 2 == 0:
                              result = result + "</tr>\n<tr>\n"
                  if item_count == 1:
                              result = result + "<td width=\"225\"
height=\"300\">\n<center><strong>" \
                                                                 + i[-1]
+ "</strong></center>\n"
                              result = result + "<br>\n<center><img
src=\"" +\
                                                                 i[-1] +
"\" height=\"177\" width=\"124\"></center>\n</br>\n"
                              result = result + "<br>\n<center>" + "Our
Price: $" +\
                                                                 i[-1] +
"<small>AUD</small>" "\n</br></center>\n</td>\n"
```

```
      else:
            result = result + "<td width=\"225\"
height=\"300\">\n<center><strong>" + \
                          str(Purchased_Items[0][0]) +
"</strong></center>\n"
            result = result + "<br>\n<center><img src=\"" \
                          + str(Purchased_Items[0][1]) + "\"
height=\"177\" width=\"124\"></center>\n</br>\n"
            result = result + "<br>\n<center>" + "Our Price: $" \
                          + str(Purchased_Items[0][2]) +
"<small>AUD</small>" "\n</br></center>\n</td>\n"
      message1_2 = result
      return message1_2

def html():
#Write into the file
      if len(Purchased_Items) > 0:
            file_name = open('invoice.html','w')
            file_name.write(message1_1 + price() + message1_2a + tables()
+ message1_3)
            file_name.close()
      else:
            file_name = open('invoice.html','w')
            file_name.write(message2)
            file_name.close()
##<---------------END OF ADDING ITEMS INTO HTML DOC--------------->

##<--------------------------EXECUTION---------------------------->
def price():
      result1 = ""
      Cost = []
      Total = 0.00
      Cost = [i[-1] for i in Purchased_Items]
      Cost =  sum(float(i) for i in Cost)
      result1 = str(float(Cost))
      return result1

def pressed():
      Save_button['state'] = 'disabled' #Locks the "Save order" button

#Reset lists
      del Item_from_one[:]
      del Item_from_two[:]
      del Item_from_three[:]
      del Purchased_Items[:]
      del Price_Item[:]
      del Cost[:]

#Timer for text being displayed on GUI
      timer = range(20)
      Timer2 = range(20)

#Call function to write html and change state of tkinter widgets
      if Moviequantity.get() > 0: #Sees if there is any value in the
Movie spinbox
            for i in range(int(Moviequantity.get())):
                  site_one()
            process1['fg'] = 'Red'
            for each in Timer2:
                  process1['text'] = "DOWNLOADING Movies ... "
                  Shopping_window.update()
      else:
            pass
      if Gamesquantity.get() > 0: #Sees if there is any value in the
Games spinbox
            for i in range(int(Gamesquantity.get())):
                  site_two()
            for each in timer:
                  process1['fg'] = 'Red'
                  process1['text'] = "DOWNLOADING Games ... "
                  Shopping_window.update()
      # else:
      #     pass
      if Musicquantity.get() > 0:  #Sees if there is any value in the
Music spinbox
            for i in range(int(Musicquantity.get())):
                  site_three()
            for each in timer:
                  process1['fg'] = 'Red'
                  process1['text'] = "DOWNLOADING Music ... "
                  Shopping_window.update()
      else:
            pass
      process1['text'] = 'Done!' #Changes step 3's message in the GUI
      Save_button['state'] = 'normal' #Unlocks the "Save order" button
after\
                #progress message says "done!"

#Call other functions
      price() #Calls the function "price"
      html() #Calls the function "html"

##<----------------------------SQL---------------------------------->

def database():
#Data to be stored
      write_me1 = Purchased_Items
      write_me2 = Purchased_Items[-1]
      for img in Purchased_Items:
            del img[1] #Deletes the image, image not needed in SQL
database

#Write SQL statements to text file
      text_out = open('Purchases.txt', 'w')
      text_out.write('DROP TABLE IF EXISTS `Purchases`;CREATE TABLE
`Purchases` ('
                  + '`Description` TEXT,`Price`     INTEGER);INSERT INTO
Purchases'
                  +'(Description, Price) VALUES')
      if len(Purchased_Items) > 2:
            for each in write_me1[0:-1]:
                  text_out.write('(\'' + each[0] + '\', \'' + each[1] +
'\'),')
            text_out.write('(\'' + write_me2[0] + '\', \'' + write_me2[1]
+ '\')')
      if len(Purchased_Items) == 2:
```

```
                 text_out.write('(\'' + write_me1[0][0] + '\', \'' +
write_me1[0][1] + '\'),')
                 text_out.write('(\'' + write_me1[1][0] + '\', \'' +
write_me1[1][1] + '\')')
      if len(Purchased_Items) == 1:
            text_out.write('(\'' + write_me1[0][0] + '\', \'' +
write_me1[0][1] + '\')')
      text_out.close()

#Execute SQL statements
      database = connect('Purchases.db')
      sqlquery = open('Purchases.txt').read()
      execute1 = findall('DROP.*?;', sqlquery)
      execute2 = findall('CREATE.*?;', sqlquery)
      execute3 = findall("INSE.*", sqlquery)
      i = database.cursor()
      i.execute(execute1[0])
      i.execute(execute2[0])
      i.execute(execute3[0])
      database.commit()
      database.close()

#Change the label under step 4
      process1['fg'] = 'Red'
      process1['text'] = "Order Saved!"
      Save_button['state'] = 'disabled' #disables the button, user must
press \
      #"print invoice" button again to unlock "save order" button


##<--------------------------END OF SQL--------------------------->

##<--------------------GRAPHICAL USER INTERFACE-------------------->

#Create a window
Shopping_window = Tk()

#Give the window a title
Shopping_window.title('Your Entertainment World Online Shop')

#Set the window size
Shopping_window.geometry('400x600')

#Add a label
label = "Welcome to 'Entertainment World' Online Shopping!\n"
the_label = Label(Shopping_window, wraplength = 380, text = label, fg =
'blue', \
                  font = ('Calibri', 30, "bold"), compound = TOP)

#Step 1, Choose your quantities
titleTK1 = "Step 1. Choose your quantities\n"
text1 = Label(Shopping_window, text = titleTK1, fg = 'red', \
              font = ('Calibri', 20, "bold"), compound = TOP)
Text_for_site1 = Label(Shopping_window, text = "Movies", fg = 'black', \
                       font = ('Calibri', 10))
Text_for_site2 = Label(Shopping_window, text = "Games", fg = 'black', \
                       font = ('Calibri', 10))
Text_for_site3 = Label(Shopping_window, text = "Music", fg = 'black', \
                       font = ('Calibri', 10))
# Text_for_site1.insert(INSERT, "Movie DVDs")
# Text_for_site1.config(font = ('Calibri', 10))
Moviequantity = Spinbox(Shopping_window, from_= 0, to = 5, width = "3")
Gamesquantity = Spinbox(Shopping_window, from_= 0, to = 5, width = "3")
Musicquantity = Spinbox(Shopping_window, from_= 0, to = 5, width = "3")

#Step 2, When ready choose your invoice
titleTK2 = "Step 2. When ready choose your invoice"
text2 = Label(Shopping_window, text = titleTK2, fg = '#DFBC30', \
              font = ('Calibri', 20, "bold"), compound = TOP)
Invoice_button = Button(Shopping_window, text = "Print Invoice", command
= pressed)

#Step 3, Watch your order's progress
titleTK3 = "\nStep 3. Watch your order's progress\n"
text3 = Label(Shopping_window, text = titleTK3, fg = '#1BAC22', \
              font = ('Calibri', 20, "bold"), compound = TOP)
process1 = Label(Shopping_window, text = "", fg = 'White', \
                 font = ('Calibri', 18), compound = TOP)

#Step 4, Save your order
TitleTK4 = "Step 4. Save your order\n"
text4 = Label(Shopping_window, text = TitleTK4, fg =  'Blue', \
              font = ('Calibri', 20, "bold"), compound = TOP)
Save_button = Button(Shopping_window, text = "Save Order", \
                     command = database, state = DISABLED)

#Pack widgets into window
the_label.grid(columnspan = 5, row = 0, column = 3)
text1.grid(columnspan = 5, row = 1, column = 3)
Text_for_site1.grid(columnspan = 5, row = 2, column = 0)
Moviequantity.grid(columnspan = 5, row = 2, column = 1)
Text_for_site2.grid(columnspan = 5, row = 2, column = 3)
Gamesquantity.grid(columnspan = 5, row = 2, column = 4, padx = (4, 0))
Text_for_site3.grid(columnspan = 5, row = 3, column = 0)
Musicquantity.grid(columnspan = 5, row = 3, column = 1)
text2.grid(columnspan = 5, row = 4, column = 3)
Invoice_button.grid(columnspan = 5, row = 5, column = 3)
text3.grid(columnspan = 5, row = 6, column = 3)
process1.grid(columnspan = 5, row = 7, column = 3)
text4.grid(columnspan = 5, row = 8, column = 3)
Save_button.grid(columnspan =  5, row = 9, column = 3)

#Start the event loop to react to user inputs
Shopping_window.mainloop()

##<---------------END OF GRAPHICAL USER INTERFACE---------------->

file_name = 'invoice.html'
# Name of the invoice file. To simplify marking, your program should
# produce its results using this file name.

file_name2 = 'Purchases.db'
#Name of Database file
```