```
# Project:      HW03 (VoMikeHW03SecHY02Ver02.py)
# Name:         Mike Vo
# Date:         02/21/17
# Description:  This program helps Konditorei Coffee Shop customers order
their coffee

# Standardize a string of text in to a title with one space character
between each word
def StandardizeTitle(strTypeName):

    lstSplittedTypeName = strTypeName.split()
    strResult = " ".join(lstSplittedTypeName)

    return strResult.title()

# Remove any blankspace and convert all characters to lowercase (a ONE
WORD string)
def SimplifyCoffeeTypeName(strTypeName):

    lstSplittedTypeName = strTypeName.split()
    strResult = "".join(lstSplittedTypeName)

    return strResult.lower()

# Get Coffee type index from a simplified (see above) ONE WORD string
def GetCoffeeTypeIndex(strTypeName):

    lstConstTypeData = [
        "jonestownbrew",
        "plymouthjolt",
    ]

    intConstErrorCode = -1

    strType = SimplifyCoffeeTypeName(strTypeName)
    for intIndex in range(len(lstConstTypeData)):
        if strType == lstConstTypeData[intIndex]:
            return intIndex

    return intConstErrorCode

# Calculate the cost of fltAmountPounds of coffee, given its type index
def GetCoffeeCost(fltAmountPounds, intTypeIndex):

    lstConstPriceData = [
        # Jonestown Brew
        10.50,
        # Plymouth Jolt
        16.95,
    ]

    return fltAmountPounds * lstConstPriceData[intTypeIndex]

# Calculate the Shipping and Handling Cost
```

```
def GetShippingPriceWithHandling(fltAmountPounds):

    fltConstShippingRate = .93
    fltConstHandlingCost = 2.50
    return fltAmountPounds * fltConstShippingRate + fltConstHandlingCost

# From state name return its tax rate in this form:
# [intTaxRate(%), strStateAbrv1, strStateAbrv2, ...]
def GetTaxRate(strState):

    lstStateTaxData = [
        7,
        [9, "WA", "WASHINGTON"],
        [9, "CA", "CALIFORNIA"],
        [9, "TX", "TEXAS"],
        [0, "OR", "OREGON"],
        [0, "FL", "FLORIDA"],
    ]

    for intIndex in range(1, len(lstStateTaxData)):
        if strState in lstStateTaxData[intIndex]:
            return lstStateTaxData[intIndex][0]

    return lstStateTaxData[0]

# From the Delivery method index, return the delivery cost and its name
in the form of
# [fltCost, strName]
# if incorrect method index then return []
def GetDeliveryCostData(intMethodIndex):

    lstErrorCode = []

    lstConstMethodCostData = [
        [20.00, "Overnight"],
        [13.00, "2-Day"],
        [0.00, "Standard"],
    ]

    if intMethodIndex in range(len(lstConstMethodCostData)):
        return lstConstMethodCostData[intMethodIndex]
    else:
        return []

# From the Payment option index, return a fee/discount Sub-total
percentage modifier (int)
# in the form of list [intMod, strinfo]
# if incorrect option index then return []
def GetPaymentOptionModifierData(intOptionIndex):

    lstErrorCode = []

    lstConstOptionModifierData = [
        [+3, "PayPal payment fee 3% of Sub-total"],
        [+5, "Credit Card payment fee 5% of Sub-total"],
        [-2, "Check payment discount 2% of Sub-total"],
    ]

    if intOptionIndex in range(len(lstConstOptionModifierData)):
        return lstConstOptionModifierData[intOptionIndex]
    else:
        return lstErrorCode

# From the amount of coffee and the coffee type index, calculate and
return a list of data
# in this format:
#   lstSubTotalData [
#       fltSubTotal = ...
#       fltCoffeeCost = ...
#       fltShippingAndHandling = ...
# ]
def GetSubTotalData(fltCoffeeAmountPounds, intCoffeeTypeIndex):

    fltCoffeeCost = GetCoffeeCost(fltCoffeeAmountPounds,
intCoffeeTypeIndex)
    fltShippingPriceWithHandling =
GetShippingPriceWithHandling(fltCoffeeAmountPounds)
    return [
        # The Sub-total
        fltCoffeeCost + fltShippingPriceWithHandling,
        # The Coffee Cost
        fltCoffeeCost,
        # The Shipping and Handling Cost
        fltShippingPriceWithHandling,
    ]


def main():

    # Print main header to shell
    print("#######################################")
    print("#      Mike Vo - CSC110 HY02 HW3      #")
    print("#               Ver. 01               #")
    print("#######################################")

    # Print coffee shop header to shell
    print("\nKONDITOREI COFFEE SHOP")
    print("Price: Jonestown Brew   $10.50/lb")
    print("       Plymouth Jolt    $16.95/lb")
    print("Shipping: $0.93/lb + $2.50 handling (fixed)")

    try:

        # Prompt user for order info from shell
        print("\nEnter your order:")
        strCoffeeTypeName = StandardizeTitle(str(input("Coffee Type>>> 
")))
        fltCoffeeAmountPounds = float(input("Coffee Amount (lbs)>>> "))
        strCity = StandardizeTitle(str(input("City>>> ")))
        strState = StandardizeTitle(str(input("State>>> ")))
        print("\nChoose one delivery method:")
        print(" 1 - Overnight: $20.00")
        print(" 2 - 2-Day: $13.00")
        print(" 3 - Standard: Free")
        intDeliveryMethodIndex = int(input(">>> ")) - 1
        print("\nChoose one payment option:")
        print(" 1 - PayPal")
        print(" 2 - Credit Card")
        print(" 3 - Check (2% discount)")
        intPaymentOptionIndex = int(input(">>> ")) - 1
        print()

        # Determine Coffee type, delivery cost data, and payment option
modifier data
        intCoffeeTypeIndex = GetCoffeeTypeIndex(strCoffeeTypeName)
        lstDeliveryCostData = GetDeliveryCostData(intDeliveryMethodIndex)
        lstPaymentOptionModifierData =
GetPaymentOptionModifierData(intPaymentOptionIndex)

        # Check for incorrect coffee type, delivery method, and payment
option
        blnRun = True
        if intCoffeeTypeIndex == -1:
            print("Can't find \"" + strCoffeeTypeName + "\" coffee
brand")
            blnRun = False
        if lstDeliveryCostData == []:
            print("Incorrect delivery method input")
            blnRun = False
        if lstPaymentOptionModifierData == []:
            print("Incorrect payment option input")
            blnRun = False

        # If determined coffee type, correct delivery method index, and
correct payment option index
        if blnRun == True:

            # Determine Sub-total, Tax, Delivery cost, Payment
fee/discount, and Total
            lstSubTotalData = GetSubTotalData(fltCoffeeAmountPounds,
intCoffeeTypeIndex)
            fltTaxRate = GetTaxRate(strState) / 100
            fltPaymentOptionModifier = lstSubTotalData[0] *
(lstPaymentOptionModifierData[0] / 100)
            fltTotal = (lstSubTotalData[0] * (1 + fltTaxRate) +
lstDeliveryCostData[0]) + fltPaymentOptionModifier

            # Print output to shell
            print(lstDeliveryCostData[1] + " delivery to\n" +
strCity.upper() + ", " + strState.upper())
            print(strCoffeeTypeName + "
({:0.2f}lbs)".format(fltCoffeeAmountPounds))
```

```
            print("-----------------------------------------")
            print("Cost                          :
${:0.2f}".format(lstSubTotalData[1]))
            print("Shipping + Handling           :
${:0.2f}".format(lstSubTotalData[2]))
            print("-----------------------------------------")
            print("Sub-total                     :
${:0.2f}".format(lstSubTotalData[0]))
            print("Delivery cost                 :
${:0.2f}".format(lstDeliveryCostData[0]))
            print("Tax                           :        " +
str(GetTaxRate(strState)) + "%")
            print("Payment*                      :
${:0.2f}".format(fltPaymentOptionModifier))
            print("-----------------------------------------")
            print("Total                         :
${:0.2f}".format(fltTotal))
            print("*" + lstPaymentOptionModifierData[1])
            print("\nThank you for shopping at Konditorei Coffee Shop :-
)")

    # Catch bad input
    except ValueError:
        print("\nWrong format input\nProgram terminated")

    # Catch other errors
    except:
        print("\nUnexpected error\nProgram terminated")

main()
```