Python Worksheet 2

1. Write a program to calculate and display the total from the bill below.

   5 x mars bars @ €1 each
   4 x cans of coke @ €1.50 each
   3 x bags of crisps @ 80 cents each
   2 x cups of tea @ €2 euro each
   1 x slice pan @ €3.50 each

   Hint: Just re-arrange the lines of code shown here.

```
costOfCoke = 4 * 1.5
costOfCrisps = qtyCrisps * unitCostOfCrisps
print("The total cost is", total)
qtyCrisps = 3
costOfBread = 3.50
qtyMars = 5
total = costOfMars+costOfCoke+costOfCrisps+costOfTea+costOfBread
costOfTea = 2 * 2
unitCostOfCrisps = 0.8
costOfMars = qtyMars * 1
```

2. Re-arrange the jumbled up lines shown below so that the program displays the sum of two integers entered by the end-user.

   Warning! There are *three* extra lines that you won't need.

```
            number2 = int(number2)
number1 = int(input("Enter first number: "))
              sum = sum + number1
            number1 = int(number1)
  print(number1, "+", number2, "=", sum)
 number2 = input("Enter second number: ")
        print("The answer is sum")
          sum = number1 + number2
```

Running Totals

Running totals are needed so often in programs that it is well worth putting some effort into understanding the pattern used to create them.

A running total is a value that usually starts at zero and increases by successive additions until a final total is reached. A very common example is a shopping basket total such as those calculated at a checkout.

Let's say we have three items in our basket and they are valued at €10, €14 and €6 respectively. With very little effort, most people understand that the total bill is €30. However, what most people probably don't realise is that they have (subconsciously) run a running total program similar to that shown below in their own heads.

```
1.  # Program to calculate a running total
2.
3.  # Initialise the variable
4.  runningTotal = 0
5.
6.  # Perform the calculations
7.  price1 = 10
8.  runningTotal = runningTotal + price1
9.  price2 = 14
10. runningTotal = runningTotal + price2
11. price3 = 6
12. runningTotal = runningTotal + price3
13.
14. # Display the output
15. print("Total amount is", runningTotal)
```

Can you re-order the lines in the previous listing without breaking the code?

Introducing Random Numbers

Random numbers provide a rich way of generating numeric data in early stage programming. A more advanced application of random number generation is in games programming.

The following two programs demonstrate random number generation.

The first multiplies two randomly generated numbers and displays the result; the second computes the mean of five randomly generated numbers.

```
1. # Program to multiply two randomly generated numbers
2. import random
3.
4. num1 = random.randint(1,10) # generate a number between 1 and 10
5. num2 = random.randint(1,10) # generate a number between 1 and 10
6.
7. # Multiply the two numbers and display the result
8. print(num1, "times", num2, "=", num1*num2)
```

```
1.  # Program to average five randomly generated numbers
2.  import random
3.
4.  low  = random.randint(1,100)
5.  high = random.randint(low,100)
6.
7.  # Generate the 5 random numbers between low and high
8.  n1  = random.randint(low, high)
9.  n2  = random.randint(low, high)
10. n3  = random.randint(low, high)
11. n4  = random.randint(low, high)
12. n5  = random.randint(low, high)
13.
14. # Compute their average
15. average = (n1+n2+n3+n4+n5)/5
16.
17. # Add the five numbers and display the result
18. print("The average of", n1, n2, n3, n4, n5, "is", average)
```

Study both programs carefully and answer the questions on the next page in relation to the second program listing.

Explain the purpose of the variables `low` and `high`.

Why do you think the variable `low` is used on line 5?

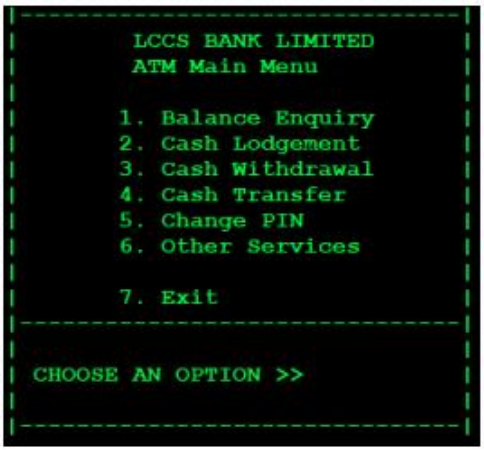
Explain why the brackets are necessary on line 15

Can you recognise how the running total pattern could be used in this program?

ATM Menu

Recall from Section 1 the menu system for our fictional LCCS bank.

Use the knowledge you have gained so far to convert the pseudo-code shown to the right of the menu below into Python.



Display a welcome message
Initialise a variable called `balance` *to* 123.45
Display the value of `balance`
Prompt the user to enter an amount to lodge
Increase the `balance` *by the amount entered*
Display the value of `balance`
Prompt the user to enter an amount to withdraw
Decrease the `balance` *by the amount entered*
Display the value of `balance`

Hint: You will need to consider what variables you will need as well as their datatype.