1.7 Examples of Programs Using the input() statement

Making More General Programs

In all of our examples, we've only been able to write programs that made very specific calculations. For example, we only found the area of one specific square, or the price of one specific item with tax. This isn't terribly helpful because not all of the items we buy will be the same price.

It would be nice if the same program can answer any price question or any square area question. This is where user input helps us. Instead of being forced to set a variable to one specific value, we can simply ask the user to enter a value, and then set a variable to that value, allowing the user to dictate the specific calculation that occurs. Here is the program that calculates the area of a square edited to allow for user input:

```
# Arup Guha
# 6/22/2012
# Python Program to calculate the area of a square - using user input.

side = int(input("Please enter the side of your square.\n"))
area = side*side
print("The area of a square with side",side,"is",area)
```

In converting our item price program to incorporate user input, we find that the type of information we are reading in is different than an integer. It's a real number, which is called a "float" in python. In order to convert a string to a float, the float function must be used. Thus, our edited program is as follows:

```
# Arup Guha
# 6/22/2012
# Python Program to cost of an item with tax.
# Edited to take in user input. Uses Orange County's sales tax rate.

tax_rate = 6.5

item_price = float(input("Please enter the price of your item.\n"))
total_price = item_price*(1+tax_rate/100)
print("Your total cost is $",total_price,".",sep="")
```

Temperature Conversion Example

While Americans use the Fahrenheit scale to measure temperature, many others use the Celsius scale. In this program, we'll ask the user to enter a temperature in Celsius and our program will convert that temperature to Fahrenheit. The formula for conversion is as follows:

$$F = 1.8C + 32$$

Here is the program:

```
# Arup Guha
# 6/22/2012
# Program to convert Celsius to Fahrenheit

temp_cel = float(input("Please enter the temperature in Celsius.\n"))
temp_fahr = 1.8*temp_cel + 32;
print(temp_cel,"degrees Celsius =",temp_fahr,"degrees Fahrenheit.")
```

Here is a sample of the program running:

```
>>>
Please enter the temperature in Celsius.
37
37.0 degrees Celsius = 98.60000000000001 degrees Fahrenheit.
>>>
```

Fuel Efficiency Example

Consider the following problem:

You are taking a road trip. When you fill up your gas tank (you know how many gallons your tank is), you notice the reading on your odometer. Later in the drive, you see exactly how much gas is left and the reading on the odometer. Given all of this information, we want to calculate how many more miles we can drive before having to stop for gas again. In real life, we would want to include a margin for error and stop several miles short of when our fuel would run out. But for purposes of simplicity, in this program, we'll simply calculate when we expect to run out of fuel, assuming that we drive with a constant fuel efficiency.

This problem is slightly more involved than the previous ones. Rather than immediately typing into the IDLE window, we need to sit back and think about the problem, sketching out what variables we want to use and how we will solve the problem.

After reading the problem statement, we see that we must read in the following variables from the user:

1) Initial Odometer Reading (in miles)
2) Gas Tank Size (in gallons)
3) Odometer Reading at Intermediate Point (in miles)
4) How Much Gas is Left at Intermediate Point (in gallons)

The difference in variables 3 and 1 represents the distance traveled while the difference of variables 2 and 4 represents the amount of gas used in the interim described. The division of the former by the latter will yield our fuel efficiency in miles/gallon. Since we know how many gallons of gas are left, we can multiply this by our fuel efficiency to see how much longer we can drive.

Let's take a look at the program:

```
# Arup Guha
# 6/22/2012
# Calculates the number of miles before having to refuel.

# Get all the user input.
start_odometer = int(input("What is the initial odometer reading?\n"))
gas_tank = float(input("How many gallons of gas does your tank hold\n"))
mid_odometer = int(input("What was your second odometer reading?\n"))
gas_left = float(input("How many gallons were left then?\n"))

# Calculate miles driven and gas used.
miles_driven = mid_odometer - start_odometer
gas_used = gas_tank - gas_left

# Calculate fuel efficiency and distance left to travel.
mpg = miles_driven/gas_used
distance_left = gas_left*mpg

print("You can go",distance_left,"miles before needing to refuel.")
```

Koolaid Example

You are running a Koolaid stand during the summer and want to be able to calculate how many glasses of Koolaid you need to sell to reach a profit goal for the day. There is a fixed daily rent for the stand itself and the other items you use. In addition, you have to buy materials (Koolaid mix, sugar). You know the cost of your materials per glass of Koolaid. Finally, you have a desired profit goal. The goal of this program is to calculate the minimum number of glasses of Koolaid that need to be sold to reach your profit goal. (Note: This answer has to be an integer.)

Once again, let's plan a bit before writing our program. First, let's idenfity the pieces of information we must get from the user:

1) Daily Rent (in dollars)
2) Cost of Material per glass (in cents)
3) Price you Charge Customers per glass (in cents)
4) Profit Goal (in dollars)

First, we can calculate the profit per glass of Koolaid by subtracting variable 2 from variable 3. We can then convert the profit goal PLUS the daily rent into cents, since this is our true revenue goal. (Due to the types of information involved, this will be a bit more difficult than multiplying by 100.) Finally, we can divide the two using integer division. Note: We find that this division will sometimes give us the incorrect answer. Consider the case where we need to earn 300 cents and we make a profit of 40 cents per glass. The integer division gives us 300//40 = 7, but if we were to only sell 7 glasses, we'd only make up 280 cents, instead of 300. In particular, we find that this division is always one glass too low, except for with the two values divide evenly, such as 280 and 40. Thus, we want every profit value from 281 through 320 divided by 40 to round up to 8, while 280 divided by 40 stays at 7. We can simply do this by adding an offset of 39 to the division. (Note: 39 represents one less than the profit per cup of Koolaid.)

Here's the whole program:

```
# Arup Guha
# 6/22/2012
# Koolaid Example

# Get the user input.
rent = int(input("How many dollars is the rent for your stand?\n"))
cost = int(input("How many cents do the materials cost, per glass?\n"))
price = int(input("How many cents do you charge per glass?\n"))
goal = int(input("What is your profit goal, in dollars?\n"))

# Calculate the profit per cup, in cents.
profit_per_cup = price - cost

# The total revenue we need to hit.
target = 100*(goal + rent)

# Calculate the final result, taking care of the off by one case.
num_cups = (target + profit_per_cup - 1) // profit_per_cup

# Print the final result.
print("You must sell at least",num_cups,"glasses to reach your goal.")
```

This example shows that care must be taken in correctly making calculations and that tools such as integer division and some creativity can be helpful in solving problems. (Note: An if statement can be used to make this problem easier. This will be covered in Chapter 2.)