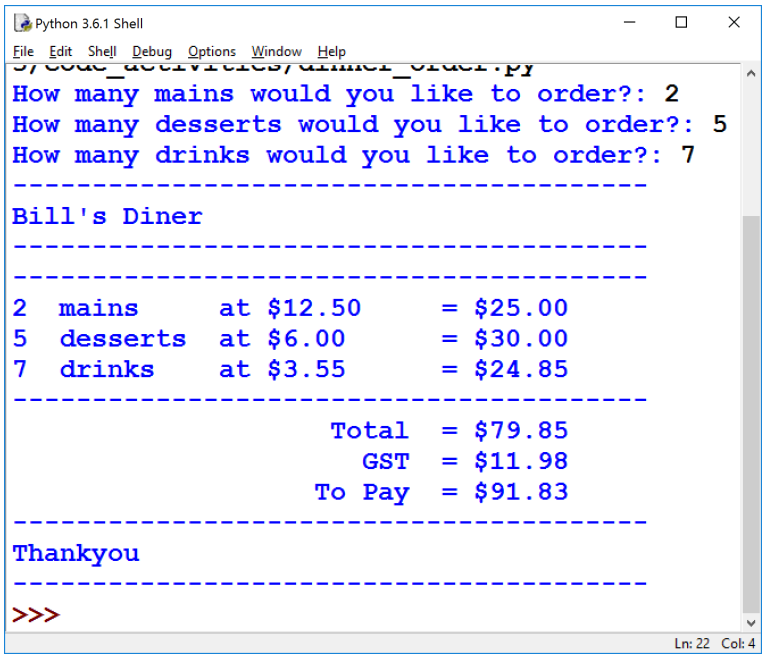
2 Basic Program

2.1 Dinner Order

You need to create a program that allows a waiter to enter in the number of main meals (at $12.50 each) , the number of desserts (at $6.00 each) and the number of drinks (at $3.55) ordered at a table.

The program then prints out a summary of the order, a total and then adds on GST (and gives the new total).

(Note: the program should be easy to alter so the price values should be declared as constant variables)



An example of Dinner Order program run

2.1.1 Planning

For the writing and planning:

- **Decomposition** Please write down your task decomposition.
  (This is breaking the task down into smaller pieces that will be combined to make the finished program)
  See examples below.

- **Version Log**
  Your version log should go here. Annotated screenshots are a good idea at this point

- **Component Trialling**
  Show that you have trialled each component here.
  You should also include notes that justify the major decisions you made.

- **Assembled Outcome Testing**

  Please show testing for your assembled outcome below.

  This should include a test plan followed by screenshot proof

- **Usability Testing**

  Try the program out on someone else and make notes or video etc. Write a list of things improvements which need to be made based on your usability testing. Then write down what you changed.

- **General Evaluation:**

  How good is may program? What can I do to improve it?

Examples of Component Planning

Basic Idea of Testing When testing we consider

Outline

3 components

Request input from the user (meal, drinks, deserts) — 1

Perform calculations
- cost of meals, drinks, deserts
- add all up
- calculate gst
- add on gst

— 2

Present a formatted output. — 3

Note: The program must use variables to store and pass information

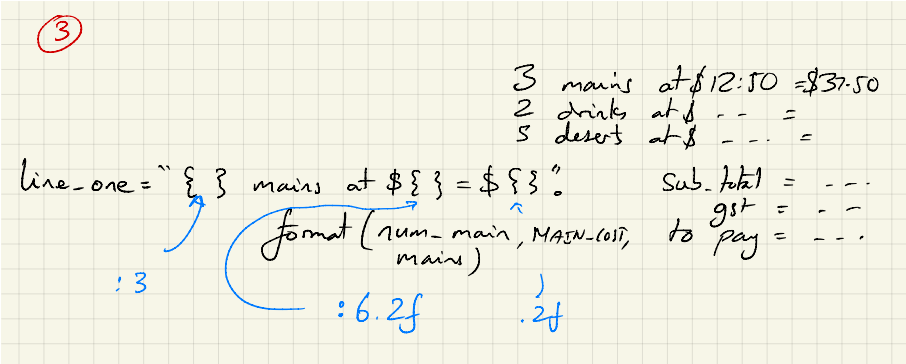
Notes.

1 3 variables to hold prices

```
MAIN_COST = 12.5   # use caps because variable is a constant value.
Same for other variables
num_main = int(input(" Please enter number of mains"))
num_desert = . . . . . .
num_drink = . . . . . .
                    # test
```

2

```
mains = MAIN_COST * num_main
deserts = DESERT_COST * num_desert
- - - -
total_cost = mains + deserts + drinks
gst = total_cost * 0.15
meal_total = total_cost + gst
                    # test
```

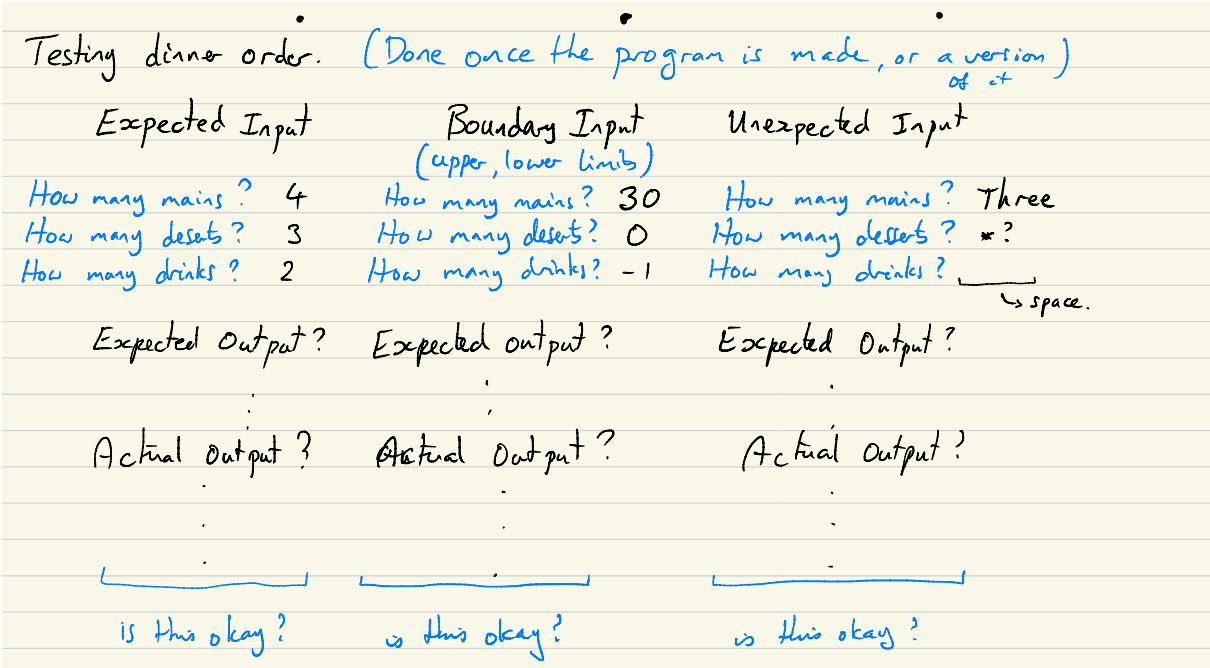


- Expected inputs (these are inputs we would normally expect the user to input).
  Does the porgram give the right outputs with the given expected inputs.

- Boundary inputs (are there maximum or minimum value that the user can enter)
  What happens at these <u>boundaries</u> (and if we go over them)

- Unexpected Inputs (these are character entries that we are not expecting)
  These could be just pressing enter or having spaces , using letters instead of numbers or characters like * , &, etc.
  These could occur if the user makes a mistake or misunderstands what is expected.

Testing should have some kind of plan and documentation of the results.



Result from expected

```
>>>
= RESTART: /Users/Paul/Desktop/code_activities/tr
How many mains would you like to order?: 4
How many desserts would you like to order?: 3
How many drinks would you like to order?: 2
----------------------------------------
Bill's Diner
----------------------------------------
4  mains     at $12.50     = $50.00
3  desserts  at $6.00      = $18.00
2  drinks    at $3.55      = $7.10
----------------------------------------
                    Total  = $75.1
                      GST  = $11.26
                   To Pay  = $86.36
----------------------------------------
Thankyou
----------------------------------------
>>> |
```

Result from boundary

```
>>>
= RESTART: /Users/Paul/Desktop/code_activities/
How many mains would you like to order?: 30
How many desserts would you like to order?: 0
How many drinks would you like to order?: -1
----------------------------------------
Bill's Diner
----------------------------------------
30 mains     at $12.50     = $375.00
0  desserts  at $6.00      = $0.00
-1 drinks    at $3.55      = $-3.55
----------------------------------------
                    Total  = $371.45
                      GST  = $55.72
                   To Pay  = $427.17
----------------------------------------
Thankyou
----------------------------------------
>>> |
```

Result from unexpected

```
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: /Users/Paul/Desktop/code_activities/trinket/dinner_order_basic.py =
How many mains would you like to order?: Three
Traceback (most recent call last):
  File "/Users/Paul/Desktop/code_activities/trinket/dinner_order_basic.py", line 1
3, in <module>
    mains = int(input("How many mains would you like to order?: "))
ValueError: invalid literal for int() with base 10: 'Three'
>>>
```

Evaluation

- What works
  - Works properly on expected inputs
  - Presentation of reciept clearly laid out on console.
- What could be improved
  - Should have an upper limit on how many meals can be entered
  - Should not allow entries below zero
  - Unexpected inputs cause a program crash
  - At the moment the program only runs once (so need a way to enter a new order)