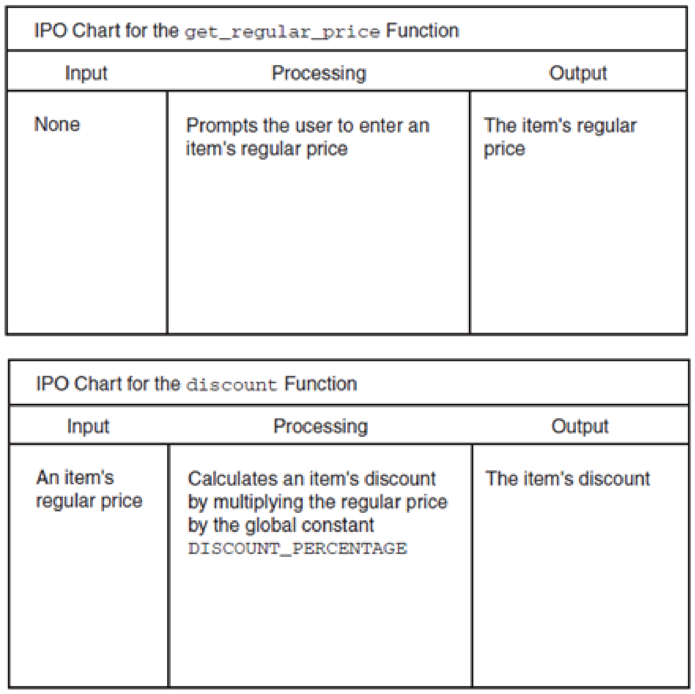
November 13, 2018

Notebook:	Computers and Programming I		
Created:	11/13/2018 2:33 PM	**Updated:**	11/13/2018 3:43 PM
Author:	Anonymous		

- Final Exam - Dec 20, 2018
- To write a value- returning function, you write a simple function and add one or more return statements
  - Format: return expression
  - The value for expression will be returned to the part of the program that called the fuction
- The expression in the return statement can be a complex expression, such as a sum of two variables or the result of another value returning fuction
- Value-returning function can be useful in specific situations
  - Example: have function prompt user for input and return the user's input
  - Simplify mathematical expression
  - Complex calculations that need to be repeated throughout the program
- Use the returned value
  - Assign it to a variable or use as an argument in another function
- IPO
  - Input Procession Output
  - Describes the input, procession, and output of a function
    - Tool for designing and documenting fuctions
    - Typically laid out in columns
    - Usually provide brief descriptions of input, processing and output without going into details
      - Often includes enough information to be used instead of a flow chart

Figure 5-25 IPO charts for the `getRegularPrice` and `discount` functions



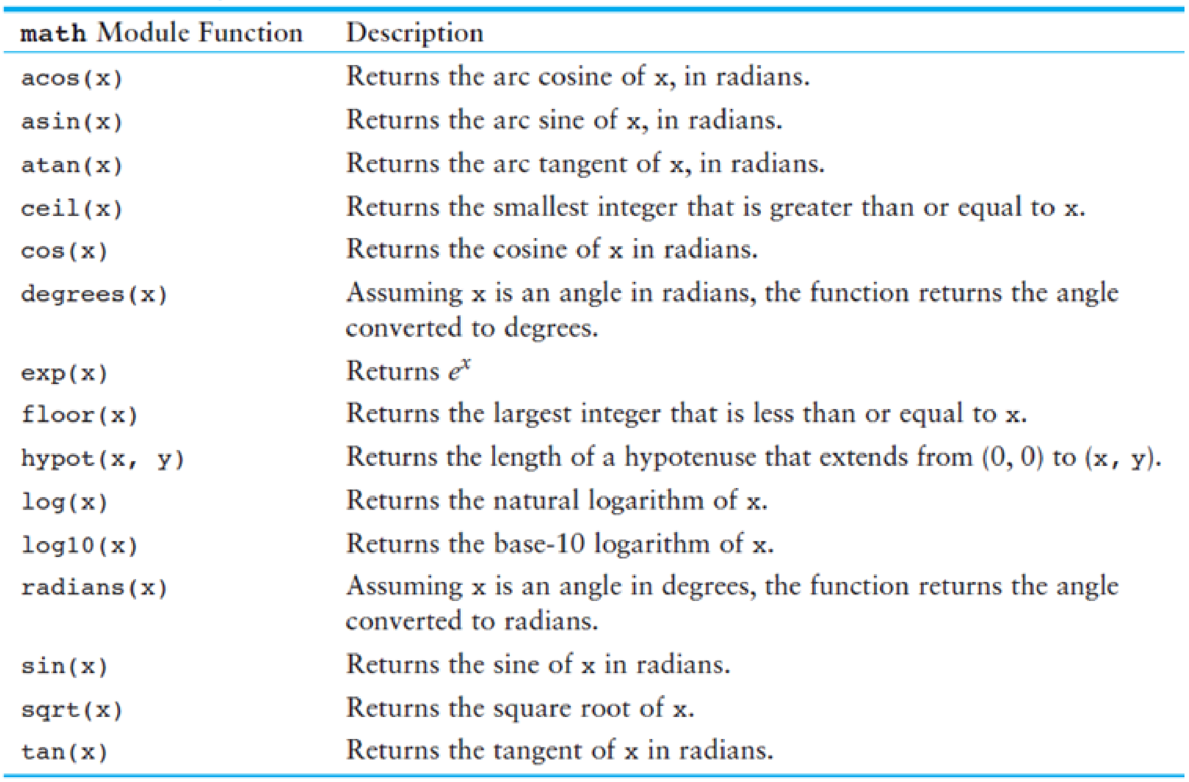
IPO Chart for the `get_regular_price` Function		
Input	Processing	Output
None	Prompts the user to enter an item's regular price	The item's regular price

IPO Chart for the `discount` Function		
Input	Processing	Output
An item's regular price	Calculates an item's discount by multiplying the regular price by the global constant `DISCOUNT_PERCENTAGE`	The item's discount

- Boolean Function:
  - Returns either true or false
    - Use to test a condition such as a for decision and repetition structures
      - Common calculations such as whether a number is even, can be easily repeated by calling a function
      - Use to simplify complex input validation code.

- Returning Multiple Arguments
    - In Python, a function can return multiple values
        - Specifies after the return statement separated by commas
            - Format: return expression1
            - expression 2, etc,
    - When you call such a function in an assignment statement, you need a separate variable on the left side of the operator = operator to receive each returned value
- The Math Module
    - Part of standard library that contains functions that are useful for performing mathematical calculations
        - Typically accept one or or values as arguments, perform mathematical operation and return the result
        - Use of module requires an import math statement

Table 5-2 Many of the functions in the `math` module



`math` Module Function	Description
`acos(x)`	Returns the arc cosine of x, in radians.
`asin(x)`	Returns the arc sine of x, in radians.
`atan(x)`	Returns the arc tangent of x, in radians.
`ceil(x)`	Returns the smallest integer that is greater than or equal to x.
`cos(x)`	Returns the cosine of x in radians.
`degrees(x)`	Assuming x is an angle in radians, the function returns the angle converted to degrees.
`exp(x)`	Returns e^x
`floor(x)`	Returns the largest integer that is less than or equal to x.
`hypot(x, y)`	Returns the length of a hypotenuse that extends from (0, 0) to (x, y).
`log(x)`	Returns the natural logarithm of x.
`log10(x)`	Returns the base-10 logarithm of x.
`radians(x)`	Assuming x is an angle in degrees, the function returns the angle converted to radians.
`sin(x)`	Returns the sine of x in radians.
`sqrt(x)`	Returns the square root of x.
`tan(x)`	Returns the tangent of x in radians.

- The math module defines variables pi and e, which are assigned the mathematical values for pi and e
    - Can be used in equations that require these values, to get more accurate results
- Variable must also be called using the dot notation
- circle_area = math.pi * radius **2
- Storing Functions in Modules
    - In large, complex programs, it is important to keep code organized
    - Modularization:
        - Grouping related functions in modules
            - Makes program easier to understant, test ans maintain
            - Make it easier to reuse code for multiple different programs
            - Import the module containing the requires function to teach program the needs it