# Chapter 5 - Peggy's Paint Job Estimator

Time required: 180 minutes

- Comment each line of code as shown in the tutorials and other code examples.
- Follow all directions carefully and accurately.
- Think of the directions as minimum requirements.

---

### Pseudocode

1. Write pseudocode for the exercise
2. Save it in a document
3. Submit with the assignment

## Requirements

Peggy has asked to write a program in Python to estimate the cost of painting a house with the following requirements.

- For every 112 square feet of wall space
    - One gallon of paint
    - Eight hours of labor
    - $35.00 per hour for labor

Write a Python program that asks the user to enter the square feet of wall space to be painted and the price of the paint per gallon.

1. Create a new Python program named **paint_job_estimator.py**
2. Import and use the **utils** module.
    a. Use the **title** function to spice up the program with a title.
    b. From the utils module, use the **get_int()** function shown in the Functions Tutorial to ask the user for the square feet of wall space and the price of the paint per gallon.
3. Use a main function as shown in the tutorials.

Please create a value returning function with these names for each of these calculations.

a. calculate_gallons_of_paint()

b. calculate_cost_of_paint()

c. calculate_hours_of_labor()

d. calculate_labor_cost()

e. calculate_total_cost()

4. Create a display function that displays the following data. This function will have each of these variables as a parameter.

   a. The number of gallons of paint required (Round up to the nearest gallon)

   b. The cost of the paint

   c. The hours of labor required

   d. The labor charges

   e. The total cost of the paint job

## TODO

The utils.py file was created in the Chapter tutorials.

```
import utils
import math
# TODO: Create Constants - something we know before the
# program starts and whose value does not change
# 112 square feet per gallon
# 8 hours of labor per gallon
# 35 dollars per hour for labor
SQUARE_FEET_PER_GALLON = 112
HOURS_PER_GALLON = 8
PER_HOUR_LABOR = 35

def main():
    # TODO: Get input from user square_feet and price of paint
    square_feet = utils.get_int("Square feet of wall space: ")
    paint_price_per_gallon = utils.get_int("Price of paint per
gallon: ")
```

```
    # TODO: Call functions and return values
    gallons_of_paint = calculate_gallons_of_paint(square_feet)

    cost_of_paint = calculate_cost_of_paint(
        paint_price_per_gallon, gallons_of_paint)

    # Display function to display all data
    display(gallons_of_paint, cost_of_paint)


# TODO: Calculate and return gallons of paint
def calculate_gallons_of_paint(square_feet):
    number_of_gallons = square_feet / SQUARE_FEET_PER_GALLON
    # Round up to the nearest integer with math.ceil()
    number_of_gallons = math.ceil(number_of_gallons)
    # Return number of gallons of paint
    return number_of_gallons


# TODO: Calculate and return cost of paint
def calculate_cost_of_paint(paint_price, gallons_of_paint):
    cost_of_paint = paint_price * gallons_of_paint
    return cost_of_paint


# TODO: Calculate and return hours of labor
def calculate_hours_of_labor(gallons_of_paint):
    pass


# TODO: Calculate and return labor cost
def calculate_labor_cost(hours_of_labor):
    pass


# TODO: Calculate and return total cost
def calculate_total_cost(labor_cost, total_cost_of_paint):
    pass
```

```
# TODO: Create a display function that displays the following data.
# This function will have each of these variables as a parameter.
# The number of gallons of paint required (Round up to the nearest
gallon)
# The hours of labor required (Round up to the nearest hour)
# The cost of the paint
# The labor charges
# The total cost of the paint job
def display(gallons_of_paint, total_cost_of_paint,  etc):
    Pass
```

Example run:

```
Square feet of wall space: 200
Price of paint per gallon: 20
Gallons of paint: 2
Hours of labor: 16
Paint charges: $40.00
Labor charges: $560.00
Total cost: $600.00
```

## Assignment Submission

1. Attach the pseudocode.
2. Attach the program files.
3. Attach screenshots showing the successful operation of the program.
4. Submit in Blackboard.