Assignment3

Part 1

Development:

1. Backward-engineer (using Python) the following screenshots:
2. The program should be organized with two modules (See Ch. 4):
    a. **functions.py** module contains the following functions:
        i. get_requirements()
        ii. estimate_painting_cost()
        iii. print_painting_estimate()
        iv. print_painting_percentage()
    b. **main.py** module imports the **functions.py** module, and calls the functions.
3. Be sure to test your program using both **IDLE** and **Visual Studio Code**.
4. ***Be Sure*** to carefully review (**How to Write Python**):
   https://realpython.com/lessons/what-pep-8-and-why-you-need-it/

Part 2

README.md file should include the following items:

1. **Assignment requirements, as per A1.**
2. Screenshot as per example below.
3. Upload A3 **.ipynb** file and create link in README.md;
   Note: *Before* uploading **.ipynb** file, ***be sure*** to do the following actions from **Kernal** menu:
    a. **Restart & Clear Output**
    b. **Restart & Run All**

Deliverables:

1. Provide **Bitbucket** read-only access to **lis4369** repo, include links to the repos you created in the above tutorials in **README.md**, using Markdown syntax (**README.md** must also include screenshots as per above.)
2. **FSU's Learning Management System**: lis4369 **Bitbucket** repo

Painting Estimator

(***Be sure*** to include *your* name as "Developer"!)

```
Painting Estimator

Program Requirements:
1. Calculate home interior paint cost (w/o primer).
2. Must use float data types.
3. Must use SQFT_PER_GALLON constant (350).
4. Must use iteration structure (aka "loop").
5. Format, right-align numbers, and round to two decimal places.
6. Create at least five functions that are called by the program:
        a. main(): calls two other functions: get_requirements() and estimate_painting_cost()
        b. get_requirements(): displays the program requirements.
        c. estimate_painting_cost(): calculates interior home painting, and calls print functions.
        d. print_painting_estimate(): displays painting costs.
        e. print_painting_percentage(): displays painting costs percentages.

Input:
Enter total interior sq ft: 2500
Enter price per gallon paint: 30
Enter hourly painting rate per sq ft: 2

Output:
Item                    Amount
Total Sq Ft:          2,500.00
Sq Ft per Gallon:       350.00
Number of Gallons:        7.14
Paint per Gallon:    $   30.00
Labor per Sq Ft:     $    2.00

Cost        Amount    Percentage
Paint:   $  214.29         4.11%
Labor:   $5,000.00        95.89%
Total:   $5,214.29       100.00%

Estimate another paint job? (y/n): y

Input:
Enter total interior sq ft: 3000
Enter price per gallon paint: 40
Enter hourly painting rate per sq ft: 2

Output:
Item                    Amount
Total Sq Ft:          3,000.00
Sq Ft per Gallon:       350.00
Number of Gallons:        8.57
Paint per Gallon:    $   40.00
Labor per Sq Ft:     $    2.00

Cost        Amount    Percentage
Paint:   $  342.86         5.41%
Labor:   $6,000.00        94.59%
Total:   $6,342.86       100.00%

Estimate another paint job? (y/n): n

Thank you for using our Painting Estimator!
Please see our web site: http://www.mysite.com
```

Part 3
Questions (Python: Ch. 6):

1. Consider the following code:

```
def main():
    students = [["Lizzy", 73, "C"],
                ["Mike", 98, "A"],
                ["Joel", 88, "B+"],
                ["Anne", 93, "A"]]

    for student in students:
        for item in student:
            print(item, end=" ")

if __name__ == "__main__":
    main()
```

What is the value of students[2][1]?
c. Joel
d. 88
e. Joel, 88, B+
f. B+

2. Consider the following code:

```
def main():
    students = [["Lizzy", 73, "C"],
                ["Mike", 98, "A"],
                ["Joel", 88, "B+"],
                ["Anne", 93, "A"]]

    for student in students:
        for item in student:
            print(item, end=" ")

if __name__ == "__main__":
    main()
```

What is in the second row of the students list?
a. "Mike", 98, "A"
b. "Joel", 88, "B+"
c. 73, 98, 88, 93
d. "C", "A", "B+", "A"

3. Consider the following code:

```
def main():
    students = [["Lizzy", 73, "C"],
                ["Mike", 98, "A"],
                ["Joel", 88, "B+"],
                ["Anne", 93, "A"]]

    for student in students:
        for item in student:
            print(item, end=" ")

if __name__ == "__main__":
    main()
```

What will display after the code executes?

a.
Lizzy,73,C
Mike,98,A
Joel,88,B+
Anne,93,A

b.
Lizzy 73 C
Mike 98 A
Joel 88 B+
Anne 93 A

c. Lizzy,73,C Mike,98,A Joel,88,B+ Anne,93,A
d. Lizzy 73 C Mike 98 A Joel 88 B+ Anne 93 A

4. Consider the following code:

```
def main():
    students = [["Lizzy", 73, "C"],
                ["Mike", 98, "A"],
                ["Joel", 88, "B+"],
                ["Anne", 93, "A"]]

    for student in students:
        for item in student:
            print(item, end=" ")

if __name__ == "__main__":
    main()
```

What would display if the following three lines were added at the end of the main() function?

students.sort()
students.reverse()
print(students)

a. [['Mike',98,'A'],['Lizzy',73,'C'],['Joel',88,'B+'],['Anne',93,'A']]
b. [['Anne',93,'A'],['Joel',88,'B+'],['Lizzy',73,'C'],['Mike',98,'A']]
c. 'Mike',98,'A', 'Lizzy',73,'C', 'Joel',88,'B+', 'Anne',93,'A'
d. [`Mike’], [`Lizzy’], [`Joel’], [`Anne’]

5. Given the following list, what is the value of ages[5]?

ages = [22, 35, 24, 17, 28]
a. 22
b. 17
c. 28
d. None: Index error

6. Given the following list, what is the value of names[2]?

names = ["Lizzy", "Mike", "Joel", "Anne", "Donald Duck"]
a. Mike
b. Joel
c. Anne
d. None, improper assignment of "Donald Duck" due to space in the name

7. Given the following code, what is the value of my_name
and what does the list consist of after the second statement is executed?

```
names = ["Lizzy", "Mike", "Joel", "Anne", "Donny"]
my_name = name.pop()
```

a. my_name = "Donny", names = ["Lizzy", "Mike", "Joel", "Anne", "Donny"]
b. my_name = "Lizzy", names = ["Mike", "Joel", "Anne", "Donny"]
c. my_name = "Donny", names = ["Lizzy", "Mike", "Joel", "Anne"]
d. Error: must specify item number with the pop() method

8. Given the following code, what would the list consist of after the second statement?

ages = [22, 35, 24, 17, 28]
ages.insert(3, 4)
a. ages = [22, 35, 24, 4, 17, 28]
b. ages = [22, 35, 3, 24, 17, 28]
c. ages = [22, 35, 24, 17, 3, 28]
d. ages = [22, 35, 24, 17, 4, 28]

9. Given the following code, what would be displayed after the code executes?

```
def main():
    furry_pets = ["dog", "cat", "ferret", "hamster", "bunny"]
    feathered_pets = ["canary", "parrot", "budgie", "hawk"]
    all_pets = furry_pets + feathered_pets
    new_pets =[]
    i = 0
    for item in all_pets:
        if item[i][0] == "c":
            new_pets.append(item)
    print("The pet store sells:", all_pets)
    print("These start with the letter c:", new_pets)
```

a. The pet store sells: ["dog", "cat", "ferret", "hamster", "bunny"]
These start with the letter c: ['cat']

b. The pet store sells: ['canary', 'parrot', 'budgie', 'hawk']
These start with the letter c: ['canary']

c. The pet store sells: ['bunny', 'cat', 'dog', 'ferret', 'hamster', 'budgie', 'canary', 'hawk', 'parrot']
These start with the letter c: ['cat', 'canary']

d. The pet store sells: ['dog', 'cat', 'ferret', 'hamster', 'bunny', 'canary', 'parrot', 'budgie', 'hawk']
These start with the letter c: ['cat', 'canary']

10. Given the tuple that follows, which of the following assigns the values in the tuple to variables?

```
numbers = (22, 33, 44, 55)
```

a. for item in numbers:
item[i] = numbers[i]
b. w, x, y, z = numbers.unpack()
c.
w = numbers
x = numbers
y = numbers
z = numbers
d. w, x, y, z = numbers

11. The _____ method adds an item to the end of a list.
pop()
append()
insert()
index()

12. The primary difference between a tuple and a list is that a tuple
has a limited range
is indexed starting from 1
is mutable
is immutable

13. To insert the item "melon" after "grapes" in the following list, you would use which of these methods?

```
fruit = ["apple", "banana", "grapes", "mangos", "oranges"]
```

fruit.pop("melon", 3)
fruit.insert("melon", 3)
fruit.insert(3, "melon")
fruit.append(3, "melon")

14. To refer to an item in a list, you code the list name followed by
an index number in brackets, starting with the number 1
an index number in parentheses, starting with the number 1
an index number in brackets, starting with the number 0
an index number in parentheses starting with the number 0

15. To remove the item "mangos" from the following list, you would use which of these methods?

```
fruit = ["apple", "banana", "grapes", "mangos", "oranges"]
```

a. fruit.remove("mangos") or
fruit.pop(3)

b. fruit.remove(3) or
fruit.pop("mangos")

c. fruit = remove("mangos") or fruit = pop(3)
d. fruit = pop("mangos")

16. What is the value of the total variable after the following code executes?

```
prices = [10, 15, 12, 8]
total = 0
i = 1
while i < len(prices):
    total += prices[i]
    i += 1
print(total)
```

0
35
8

17. What will display after the following code executes?

```
def add_item(list, food):
    food = "apple pie"
    list.append(food)

def main():
    lunch = ["sandwich", "chips", "pickle"]
    food = "banana"
    add_item(lunch, food)
    print(lunch)

main()
```

a. ['sandwich', 'chips', 'pickle', 'banana']
b. ['sandwich', 'chips', 'pickle', 'banana', 'apple pie']
c. ['sandwich', 'chips', 'pickle', 'apple pie']
d. Error: list is undefined

18. What would be displayed after the following code snippet executes?

```
costumes = ["ghost", "witch", "elf", "ogre"]
name = "elf"
if name in costumes:
    costumes.remove(name)
for item in costumes:
    print(item)
```

a. ghost
witch
ogre

b. ghost, witch, ogre

c.
"ghost"
"witch"
"elf"
"ogre"

d. "ghost", "witch", "elf", "ogre"

19. When a function changes the data in a list, the changed list
does not need to be returned because lists are mutable.
is only available within that function.
needs to be returned because lists are immutable.
does not need to be returned because lists are immutable.

20. When you use a multiple assignment statement to unpack a tuple,
you assign the tuple to a list
you assign the tuple to a two or more variable names separated by commas
you use a for statement to assign the values in the tuple to a list
you use indexing to assign the values in the tuple to multiple variables

21. Which of the following is not true about a list of lists?
You can use nested for statements to loop through the items in a list of lists.
You can refer to an item in an inner list by using two indexes.
To delete an item in the outer list, you first have to delete the list in the item.
The inner lists and the outer list are mutable.

22. Which of the following functions randomly selects one item in a list?
choice()
shuffle()
lrandom()
randomitem()

23. Which of the following statements about list copies is not true? When you make a
deep copy of a list, both variables refer to their own copy of the list.
deep copy of a list, both variables refer to the same list.
shallow copy of a list, both variables refer to the same list.
shallow copy of a list, the list is immutable.

24. Which of the following would create a list named numbers consisting of 3 floating-point items?

a.
numbers[1] = 5.3
numbers[2] = 4.8
numbers[3] = 6.7
b. numbers = [5.3, 4.8, 6.7]
c. numbers = [0] * 3
d. numbers[3] = (5.3, 4.8, 6.7)

25. Which of the following creates a tuple of six strings?
vehicles = ("sedan","SUV","motorcycle","bicycle","hatchback","truck")
vehicles = ["sedan","SUV","motorcycle","bicycle","hatchback","truck"]
vehicles = (sedan, SUV, motorcycle, bicycle, hatchback, truck)
vehicles = "sedan","SUV","motorcycle","bicycle","hatchback","truck"