Activity 3 - Code Review Hands One

Code Snippets to Review

You will need to provide feedback as demonstrated in the assignment description for the following five snippets. Please use this template document to complete this assignment. You will need to provide written feedback and also “corrected” code just like in the examples in the assignment description.

Name: Darwin Yee

email: yeeda@oregonstate.edu

Snippet 1

```
# Pull request 1
def my_func(x):


    return x**2
```

Feedback: Place your feedback and your “corrected code here”

The code is precise and fits the style guide. Nevertheless, it is not clear as to what the function does. Comments are missing and the function name reveals nothing about the function. It looks like the function returns the square of x. Here is my suggested code:

```
def calculate_square(val):
    “““return the square of val”””
    return val**2
```

Snippet 2

```
# Pull request 2
def create_odds(num):
    “““Creates a list of len(num) of random odd numbers between 1 and 1000”””
    num_list = []
    for i in range(0, num):
        new_num = 2
        while new_num % 2 == 0:
            new_num = random.randint(1, 1000)
        num_list.append(new_num)
    return num_list

def create_evens(num):
    “““Creates a list of len(num) of random even numbers between 1 and 1000””
”
    num_list = []
    for i in range(0, num):
        new_num = 1
        while new_num % 2 != 0:
            new_num = random.randint(1, 1000)
        num_list.append(new_num)
    return num_list
```

Feedback: The code is well structured and follows the style guide. There is a comment in each function explaining what the function returns. Nevertheless, there are a lot of repetitive codes. This is violating the DRY rule. Also, the parameter name should be clearer so users can use the appropriate argument for these functions. The “num” should be changed to “number_of_integers”. To reduce the repeated code and to make the code more pythonic, random.randrange should be used. Here is my suggested code:

```
def create_odds_or_evens(number_of_integers, is_even):
    “““return a list of either odds or evens between 1 and 1000 inclusive
       number_of_integers is the number of integers to be generated
       is_even takes Boolean value True or False”””
    if type(is_even) is not type(True):
        raise TypeError('is_even is not boolean')
    return [random.randrange(1 + is_even, 1000 + is_even, 2)
            for i in range(number_of_integers)]
```

Snippet 3

```
# Pull request 3
def check_for_val(self, val):
    “““This member function checks to see if val exists in the class member
    values and returns True if found”””
    for i in range(len(self.values)):
        if self.values[i] == val:
            return True
    return False
```

Feedback: The code is clean and fits the coding style. A comment is included to describe the purpose of this function. Nevertheless, the function name may be a little unclear as to what the function is supposed to return. Also, the “val” parameter should be named more specific, like val_to_check or targeted_val. Finally, to be more pythonic, the for loop and the if statement can be replaced by the any() function. Here is my suggested code:

```
def check_if_val_exists(self, val_to_check):
    “““This member function checks to see if val_to_check exists in the class
    member values and returns True if found”””
    return any(item == val_to_check for item in self.values)
```

Snippet 4

```
# Pull request 4
def get_val_index(arr, val):
    “““Searches arr for val and returns the index if found, otherwise -1”””
    index = -1
    for i in range(len(arr)):
        if arr[i] == val:
            index = i
            break
    return index
```

Feedback: The code is clean and follows the style guide. The comment describes precisely what the function does. Nevertheless, the parameters should be named more specific. “arr” should be named “list_of_values” and “val” should be named “val_to_check”. To make it more pythonic, the index can be returned directly instead of storing to a separate variable and the .index function for list should be used. Here is my suggested code:

```
def get_val_index(list_of_values, val_to_check):
    “““Searches list_of_values for val_to_check and returns the index if
    found, otherwise -1”””
    return list_of_values.index(val_to_check)\
        if val_to_check in list_of_values else -1
```

Snippet 5

```
# Pull request 5
int_arr = [1, 2, 5, 2, 10, 45, 9, 100]

def print_sorted(arr):
    “““Prints the items in the array after sorting”””
    arr.sort()
    for num in arr:
        print(num)
```

Feedback: The code is clean, precise and easy to understand. The comment clearly describes what the function is intended to accomplish. However, the code is not pythonic. Here is my suggested code:

```
def print_sorted(arr):
    “““Prints the items in the array after sorting”””
    arr.sort()
    print(*arr, sep=’\n’)
```