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: Xiaoying Li

email: lixiaoyi@oregonstate.edu

Snippet 1

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


    return x**2
```

Feedback: The code works well, good job. But it doesn’t follow PEP8 style guide and has a linting error, which is too many blank lines. Also, the name of the function provides no clue what this function is meant to do. I think it would be better to have clearer function name and variable name, and comment to explain what the function is doing. No matter how simple the code is, clearer names and good comments can always improve the readability. See my suggested code below.

```
def second_power(val):

    # Returns the second power of the 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 works well, all functions and variables are named properly, and has detailed comments, good job. Did you know Python offers a function that can generate random numbers from a specified range and also allowing rooms for steps to be included, called randrange() in random module? Also, I think there is no need to declare a num_list and append all generated random numbers one by one to the list. Instead, just put all generated random numbers in a list and return the list itself would be much more succinct. Finally, there is a little linting error that two blank lines are expected between functions, but your code only has one blank line. See my suggested code below.

```
def create_odds(num):
    # Creates a list of len(num) of random odd numbers between 1 and 1000
    return [random.randrange(1, 1000, 2) for i in range(0, num)]


def create_evens(num):
    # Creates a list of len(num) of random even numbers between 1 and 1000
    return [random.randrange(2, 1001, 2) for i in range(0, num)]
```

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 works well and follows the style guide, all functions and variables are named properly, and has detailed comments, good job. Did you know Python has a membership operator “in” that returns True if a sequence with the specified value is present in the object? You can use the operator and just return if val is “in” values to make the code more succinct. See my suggested code below.

```
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
    return val 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 works well and follows the style guide, and has detailed comments, good job. But it is advisable to stay away from naming your variables the same thing as Python’s builtins. In this case, you overwrite index, which can cause issues later. Also, did you know Python has the index() method that returns the position at the first occurrence of the specified value in a list? You can wrap this method in a try-except block. Let the function try returning the index of the occurrence of val in arr, if no valid index value can be returned, return -1. See my suggested code below.

```
def get_val_index(arr, val):
    # Searches arr for val and returns the index if found, otherwise -1
    try:
        return arr.index(val)
    except ValueError:
        return -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 works well, all functions and variables are named properly, and has detailed comments, good job. Did you know Python has the sorted() function that returns a sorted list of the specified iterable object? You can use this function to sort the array and just print what is returned from the function. Finally, there is a little linting error that two blank lines are expected before def, but your code only has one blank line. See my suggested code below.

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


def print_sorted(arr):
    # Prints the items in the array after sorting
    print(sorted(arr))
```