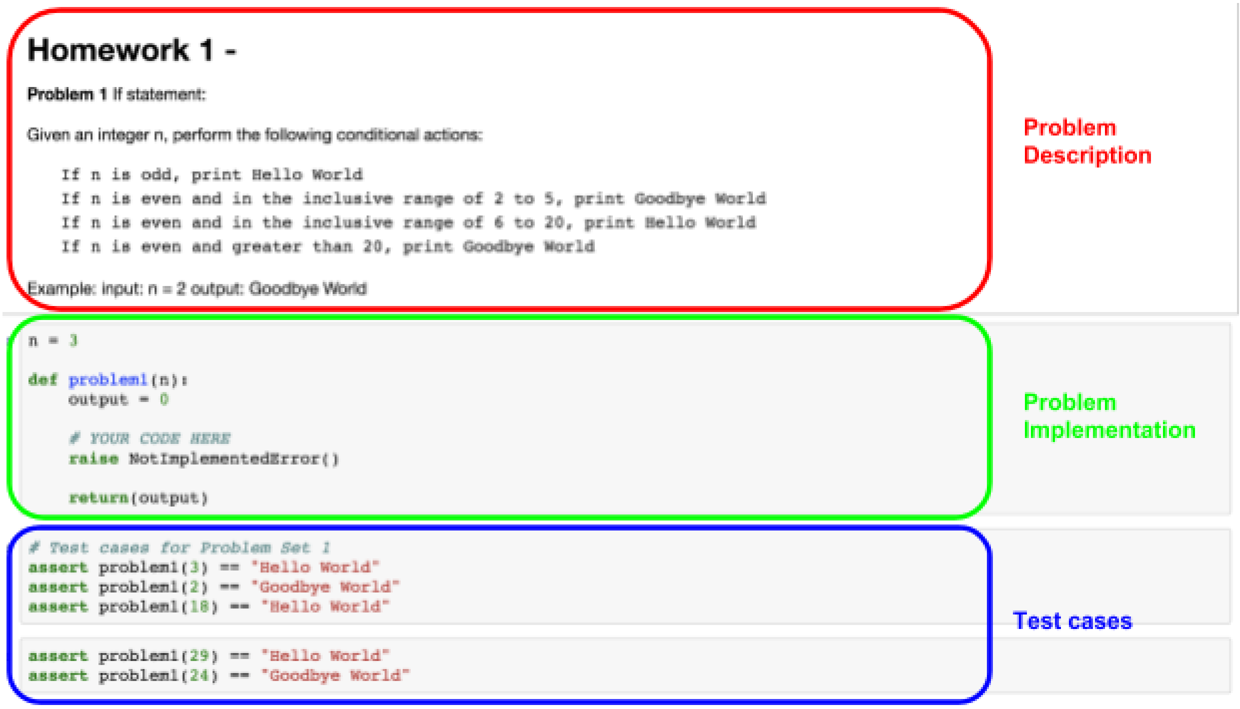
Homework Submission Guide

I. The decomposition of each homework problem

Each homework is an IPython notebook that asks you to implement certain functionalities. Each homework problem can be divided into 3 parts:

1. **The problem description:** this code block is normally in markdown format and it describes the problem.
2. **The problem implementation:** this code block is where you will enter your code to implement a particular functionality requested in the problem description.
3. **The problem test cases:** There may be several test-case code blocks. Eachcode block involves test cases to assess whether the functionality you implement in the previous block solves the problem. Note that this block may also include **HIDDEN** test cases where you cannot see on you homework. Those test cases are used to test whether you have thought about all the corner cases of your problem.

For example, the following Figure shows the three parts associated with Problem 1 in Homework 1.



Homework 1 -

Problem 1 If statement:

Given an integer n, perform the following conditional actions:

```
If n is odd, print Hello World
If n is even and in the inclusive range of 2 to 5, print Goodbye World
If n is even and in the inclusive range of 6 to 20, print Hello World
If n is even and greater than 20, print Goodbye World
```

Example: input: n = 2 output: Goodbye World

```
n = 3

def problem1(n):
    output = 0

    # YOUR CODE HERE
    raise NotImplementedError()

    return(output)
```

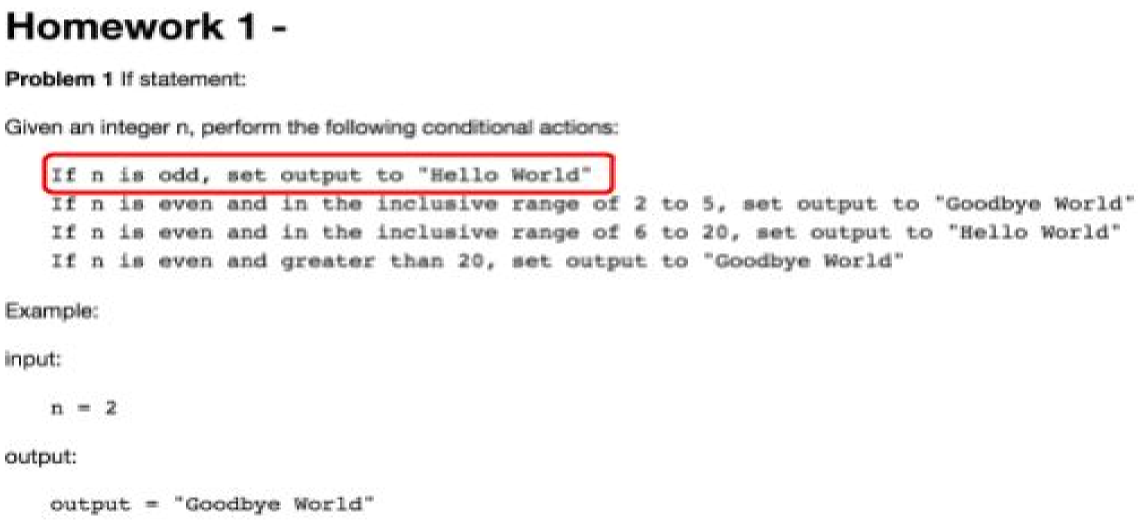
```
# Test cases for Problem Set 1
assert problem1(3) == "Hello World"
assert problem1(2) == "Goodbye World"
assert problem1(18) == "Hello World"
```

```
assert problem1(29) == "Hello World"
assert problem1(24) == "Goodbye World"
```

II. How to finish each homework problem

It takes 3 steps to finish each homework problem.

Step 1: You need to read through the problem description to understand what the functionality you are asked to implement.
For example, for Problem 1 in Homework 1, you are asked to set output equal to “Hello world” if n is odd.



Homework 1 -

Problem 1 If statement:

Given an integer n, perform the following conditional actions:

```
If n is odd, set output to "Hello World"
If n is even and in the inclusive range of 2 to 5, set output to "Goodbye World"
If n is even and in the inclusive range of 6 to 20, set output to "Hello World"
If n is even and greater than 20, set output to "Goodbye World"
```

Example:

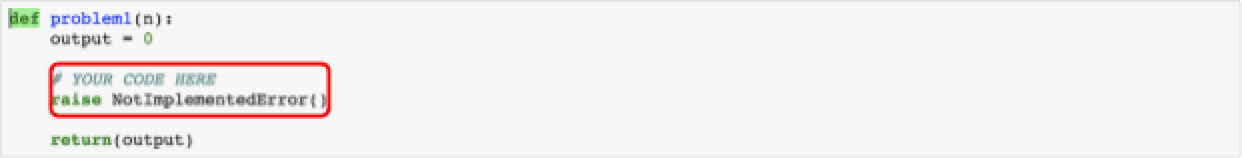
input:

```
n = 2
```

output:

```
output = "Goodbye World"
```

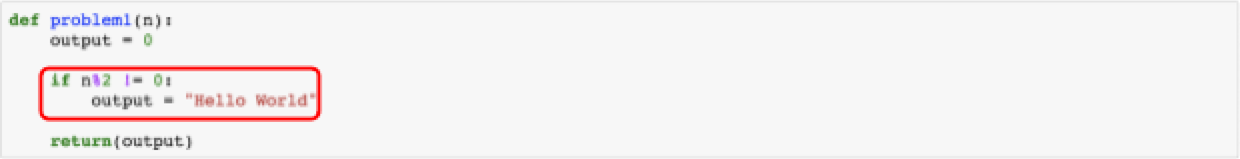
Step 2: You need to implement you code in the problem-implementation block. **You need to delete the two lines “# YOUR CODE HERE raise NotImplementedError()” and implement your code at the exact place of these two lines.**
For instance, in the following, we delete these two lines and set output to “Hello World” if it is odd.



```
def problem1(n):
    output = 0

    # YOUR CODE HERE
    raise NotImplementedError()

    return(output)
```



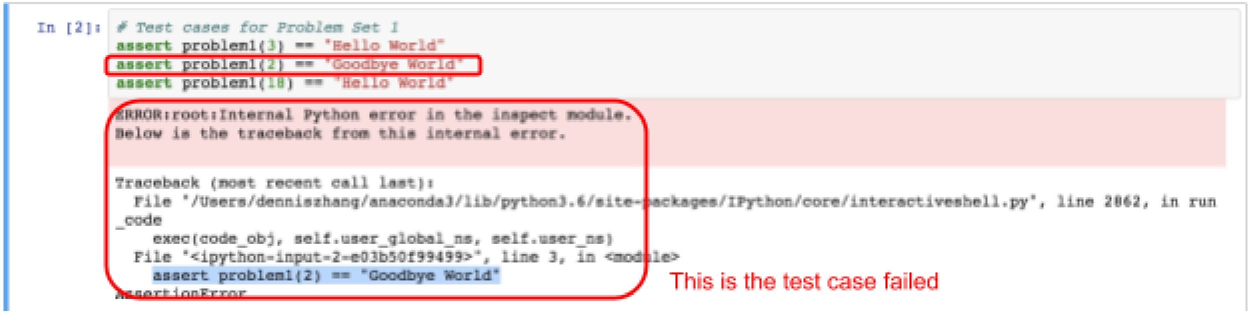
```
def problem1(n):
    output = 0

    if n%2 != 0:
        output = "Hello World"

    return(output)
```

Step 3: You need to run the test-case block to see whether your code has passed the test cases. If not, you need to fix your code in Step 2.

For example, we run the previous implementation in Step 2 and see the test case is failed. In particular, we see “assert problem1(2) == "Goodbye World"”



```
In [2]: # Test cases for Problem Set 1
        assert problem1(3) == "Hello World"
        assert problem1(2) == "Goodbye World"
        assert problem1(18) == "Hello World"

ERROR:root:Internal Python error in the inspect module.
Below is the traceback from this internal error.

Traceback (most recent call last):
  File "/Users/dennischang/anaconda3/lib/python3.6/site-packages/IPython/core/interactiveshell.py", line 2862, in run
_code
    exec(code_obj, self.user_global_ns, self.user_ns)
  File "<ipython-input-2-e03b50f99499>", line 3, in <module>
    assert problem1(2) == "Goodbye World"
AssertionError
```

This is the test case failed

Step 4: When you have implemented all the code and made sure that you passed all the test cases, you still need to make sure that your code covers all the corner cases since there may be hidden test cases.

After you finish checking all the code, you need to re-name your homework ipython notebook for grading purposes. In particular, you need to name your file as follows:

[your wustl SIS ID]_homework_[assignment number].ipynb

For example, suppose my SIS ID is 4095601, then the name for homework 1 would be:

4095601_homework_1.ipynb

Last, you can submit the ipynb notebook on Canvas assignment.