Training Task (20 minutes)

In this study, you will have the opportunity to use a new programming environment called Crowd Microservices. As you're working and find yourself stuck, you're welcome to use the Internet and any websites you'd like to help you out.

In this study, you will 1) watch and read a series of tutorials to learn how to use a new type of programming environment, 2) use the environment to complete a series of programming tasks, and 3) answer questions about your experiences working in the new programming environment.

Pleas follow tutorial steps:

Step 1:

Please watch **CAREFULLY** and **COMPLETELY** our tutorial video on YouTube: https://youtu.be/qQeYOsRaxHc

Step 2:

After watching the tutorial video please read the bellow material.

Crowd Microservices contains two types of microtasks you'll be asked to do: Implement Function Behavior and Review.

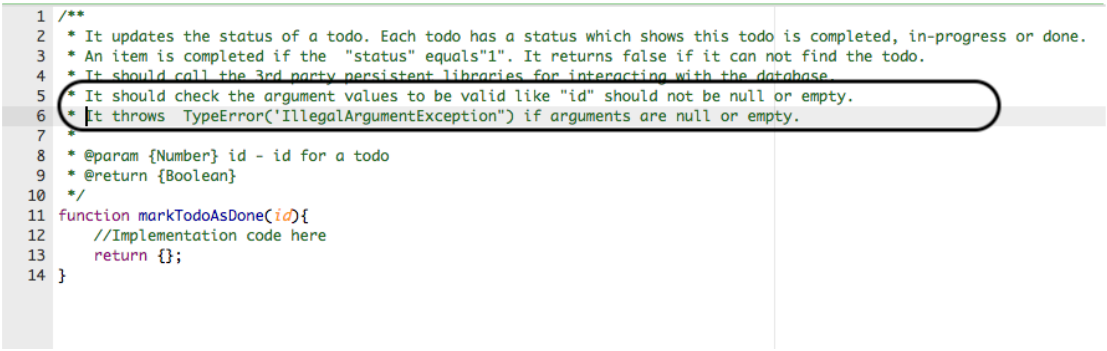
Implement Function Behavior Microtask

The Behavior Driven Development (BDD) concept is an outside-in (write a test and then code) way of development. We applied BDD to Implement Function Behavior microtask, in this kind of microtask the developer designs and writes the Unit tests for a code module before writing a single line of the code module. The developer then creates the code to pass the Unit test. Thus, this

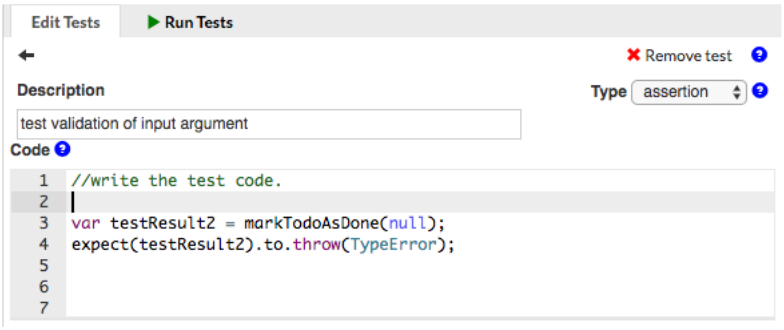
approach using unit testing to drive development. The fundamental point to note that the goal is development based on testing.

Steps in BDD:

- 1: finding a behavior − In the function description, located in the comments above the function, find one, and only one, behavior that is not yet implemented. For example: “test validation of input arguments”.



- 2: writing unit test − Write a test for the behavior you have chosen. The test fails, as the code is still not written. Hence, Step 2 is usually referred to as writing a test to fail.



- 3: Implementing function body − For the one behavior you selected, implement the behavior in the function editor. For example, to implement “test validation of input arguments”.

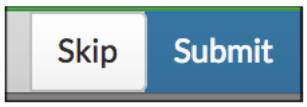
```
function markTodoAsDone(id){
    //Implementation code here

    //check validity of arguments
    if (id ===null || id ===="") {

        throw new TypeError('IllegalArgumentException');
    }

return true ;
}
```

- 4: Submit – Run all the tests to ensure that they all still pass. Unit tests are automated to facilitate this step. Submit microtask. You can submit microtask even if it is not completed.



Review Microtasks

In the Review microtask, you assess contributions submitted by other workers. You are given a diff of the code submitted with the previous version as well as the tests of the function. Instead of binary classification of rejecting or accepting microtasks, you are asked to assign a rating of 1 to 5 stars. If you evaluate the work with 1 to 3 stars, the work is marked as needing revision. You then should describe aspects of the contribution that they feel need improvement, and a microtask is generated to do this work. If you evaluate the submitted contribution with 4 or 5 stars, the contribution is accepted as is.