1) What are microtasks? What is a microtask queue? What is their role in Promises and how are they different from callbacks?

MICROTASKS

- Made of JavaScript code
- gets placed on a queue
- run at an appropriate time

1. They are scheduled so that they get executed right after the code which is being executed at that point of time
2. The additional microtasks queued during the execution of current ones are added to the end of the queue so that they get priority over others and continue to get executed
3. They include mutation observer callbacks, and promise callbacks
4. Example : make something async without taking the penalty of a whole new task.

USE: allows to schedule code to jump in front of other things in the long set of things waiting to happen on the computer

Microtasks are dominant over callback ques.The callbacks in the callback que is executed after the callbacks of the microtask queue is executed
A callback function can wait for its execution in any one of the queue, that is callback queue or microtask queue and once the call stack is empty, event loop pushes the callback functions from these queues to call stack for there execution. Microtask queue takes priority over the callback queue and callback functions of Microtask queue are executed first. Once all the functions of callback queue are executed completely, then only callback functions of callback queue gets executed

Where microtasks come into picture in promises

One situation in which microtasks can be used to ensure that the ordering of execution is always consistent is when promises are used in one clause of an if-else statement or other conditional statement, but not in the other clause.

2)Explain with examples how private, protected variables can be implemented in classes and how can they be used in subclasses?

Well to be able to answer this question, I should have proficiency in Java , which i currently do not hold ,thus excuse.
I will definitely make sure to start learning once I am taken onboard.