Q1:

A callback function associated with an async. Code waits for its execution inside either a callback queue or microtask queue. Most of the call-back associated with a async. Code, have to wait for their execution in the call-back function, for example, the call-back function associated with the promise waits for the execution in the microtask queue. The callback functions in the microtask queue, takes priority over the call-back function in the callback queue i.e., the callback functions in microtask queue get executed first.

Q2.

**Private**: are members are only accessible within the class that instantiated the object.

**Protected**: will allows a little more access than private members. A protected is accessible within the class and any object that inherits from it. A protected value is shared across all layers of the prototype chain. It is not accessible by anybody else.

Eg: Private

class ObjectCreator {

#private\_member;

constructor() {

this.#private\_member= 42;

}

returnPrivateMember() {

return this.#private\_member;

}

}

const myObject = new ObjectCreator();

console.log(myObject.returnPrivateMember()); // 42

console.log(myObject["#private\_member"]); // undefined

console.log(myObject.#private\_member); // SyntaxError

Eg: Protected

class NameGenerator {

\_name;

constructor(name) {

this.\_name = name;

}

get name() {

return this.\_name;

}

}

let nameGenerator = new NameGenerator("John");

console.log(nameGenerator.name); // John

nameGenerator.name = "Jane"; // Cannot assign to 'name'