(Notes) Prototypes

Ques 1: What will be the output of the following code?
 Explanation:
 The output will be:

Driving a vehicle
Driving a car

This is because `vehicle.drive()` invokes the `drive()` method from the **Vehicle prototype**, while `car.drive()` invokes the **overridden drive() method** in the **Car prototype**.

Ques 2: Explain the difference between `__proto__` and `prototype` in JavaScript.
 Explanation:

- `__proto__`: It points to the prototype of an object and is used for inheritance and accessing the prototype chain.

- `prototype`: It exists on constructor functions to set up inheritance for objects created by that function, defining shared properties and methods for instances.

Ques 3: What is setPrototypeOf?
 Explanation:
 `setPrototypeOf` is a method used to set the prototype (the object's internal `[[Prototype]]`) of a specified object to another object or `null`. It allows changing the prototype dynamically after an object has been created.

Ques 4: What is instanceof?
 Explanation:
 `instanceof` is an operator that checks if an object is an instance of a specific constructor or its prototype chain. It returns **true** if the object is an instance of the constructor or a constructor's prototype chain.

Ques 5: How can you create an object without a prototype in JavaScript?
 Explanation:
 You can create an object without a prototype by using:

Object.create(null)

This creates an object with no prototype chain, making it free from any inherited properties or methods.

Ques 6: What will be the output of the following code?
 Explanation:
 The output will be:

10
20
30

This is because `obj1` is an instance of `A` with `foo` as **10**, `obj2` is an instance of `B` with `foo` as **20** (overridden from `A`), and `obj3` is an instance of `C` with `foo` as **30** (overridden from both `A` and `B`).

Ques 7: Deep Clone an object in JS
 Explanation:
 The `deepClone` function recursively creates a **deep copy** of an object, including nested objects and arrays, ensuring that the cloned object is completely independent of the original object.