# JavaScript Prototype Assignment

## Section 1: Theory Questions

1. Define prototype and __proto__ in JavaScript. How are they different?
2. What happens when a property is not found on an object? Explain prototype chain lookup.
3. What is the use of Object.create() and how does it relate to prototypes?
4. How do function constructors use prototypes in JavaScript?
5. Explain how you can override a method using prototype.

## Section 2: Coding Tasks

Q1: Create a custom constructor function with prototype methods

```
function Car(make, model) {
  this.make = make;
  this.model = model;
}

Car.prototype.getDetails = function () {
  return `${this.make} ${this.model}`;
};

// Create 2 Car objects and log their details
// Extend this to add a method called startEngine via prototype.
```

Q2: Inherit using prototype chain

```
function Animal(name) {
  this.name = name;
}

Animal.prototype.speak = function () {
  return `${this.name} makes a sound`;
};
```

```
function Dog(name, breed) {
  Animal.call(this, name);
  this.breed = breed;
}

Dog.prototype = Object.create(Animal.prototype);
Dog.prototype.constructor = Dog;

// Add Dog-specific method: bark()
// Create a Dog instance and call both speak() and bark()
```

Q3: Use Object.create() to inherit

```
const person = {
  greet: function () {
    return `Hello, my name is ${this.name}`;
  },
};

const student = Object.create(person);
student.name = 'Alice';

// Log the greeting
```

Q4: Add prototype methods dynamically and show shared access

```
function Book(title) {
  this.title = title;
}

const book1 = new Book('JS Guide');
const book2 = new Book('Advanced JS');

// Dynamically add getTitle() to prototype
// Call getTitle() on both book1 and book2
```

## Section 3: Debugging and Reasoning

Q1: What's the output and why?

```
function Gadget() {}
Gadget.prototype.price = 100;

const g1 = new Gadget();
g1.price = 200;

console.log(g1.price);  // ?
delete g1.price;
console.log(g1.price);  // ?
```

Q2: Constructor reference fix

```
function Laptop() {}

Laptop.prototype = {
  brand: 'HP',
};

const l1 = new Laptop();
console.log(l1.constructor === Laptop);  // false  why? Fix it.
```

## Section 4: Advanced Tasks

Q1: Implement a clone method using prototype

```
function Person(name, age) {
  this.name = name;
  this.age = age;
}

Person.prototype.clone = function () {
  return new Person(this.name, this.age);
};
```

```
// Test cloning
```

Q2: Create a chain of inheritance

```
function LivingBeing() {}
LivingBeing.prototype.isAlive = true;

function Human() {}
Human.prototype = Object.create(LivingBeing.prototype);

function Developer(name) {
  this.name = name;
}
Developer.prototype = Object.create(Human.prototype);
Developer.prototype.code = function () {
  return `${this.name} is coding`;
};

// Create a Developer and show access to isAlive
```