**JS Constructor Exercises – Level 3**

1. Write JavaScript code to create two objects, `*parent1*` and `*child1*`. `*parent1*` should have properties `*name*`, `*age*`, and `*heritage*`. `*child1*` should inherit from `*parent1*` and have its own `*name*` and `*age*` properties. Print the values of `*child1.name*`, `*child1.age*`, and `*child1.heritage*` to the console.

* Expected Output: **Child 1's name: Dor**
* Expected Output: **Child 1's age: 7**
* Expected Output: **Child 1's heritage: Jewish**

1. Create a more complex object hierarchy in JavaScript. Define a `*grandparent*` object with properties `*name*` and `*age*`. Create a `*parent2*` object that inherits from `*grandparent*` and has an additional property `*job*`. Create a `*child2*` object that inherits from `*parent2*` and has its own property `*hobby*`. Print the values of `*child2.name*`, `*child2.age*`, `*child2.job*`, and `*child2.hobby*` to the console.
   * Expected Output: **Child 2's name: Charlie**
   * Expected Output: **Child 2's age: undefined**
   * Expected Output: **Child 2's job: Engineer**
   * Expected Output: **Child 2's hobby: Painting**
2. Implement object inheritance using constructor functions in JavaScript. Define a constructor function `*Person*` with properties `*name*` and `*age*`. Create instances of `*Person*` as `*parent3*` and `*child3*`, and set their `*name*` and `*age*` properties. Print the values of `*child3.name*` and `*child3.age*` to the console.
   * Expected Output: **Child 3's name: Bob**
   * Expected Output: **Child 3's age: 7**
3. Create JavaScript objects with object composition for inheritance. Define a common `*parent4*` object with properties `*name*` and `*age*`. Create two child objects, `*child4a*` and `*child4b*`, using `*Object.create*` and object composition. Set different `*name*` and `*age*` properties for each child, and add an extra property, `*hobby*`, to one of the children. Print the properties of both children to the console.
   * Expected Output: **Child 4a's name: Bob**
   * Expected Output: **Child 4a's age: 7**
   * Expected Output: **Child 4a's hobby: undefined**
   * Expected Output: **Child 4b's name: Charlie**
   * Expected Output: **Child 4b's age: 10**
   * Expected Output: **Child 4b's hobby: Painting**
4. Implement object inheritance using a factory function in JavaScript. Create a factory function `*createPerson*` that returns an object with properties `*name*` and `*age*`. Create instances of `*Person*` as `*parent5*` and `*child5*`, and set their `*name*` and `*age*` properties. Print the values of `*child5.name*` and `*child5.age*` to the console.
   * Expected Output: **Child 5's name: Bob**
   * Expected Output: **Child 5's age: 7**
5. Create a JavaScript object called `*circle*` with properties `*radius*` and `*area*`. Implement a method `*calculateArea*` for the `circle` object that calculates and sets the `*area*` based on the formula: `*area = π \* radius \* radius*`. After defining the `*circle*` object and its method, call `*calculateArea*` and print the `*area*` to the console.
   * Expected Output: **The circle's area is [calculated value].**
6. Define an object called `*student*` with properties `*name*`, `*age*`, and `*grades*` (an array). Add a method `*calculateAverageGrade*` to the `*student*` object that calculates and returns the average grade from the `*grades*` array. Create a `*student*` instance, add grades, and call the `*calculateAverageGrade*` method. Print the average grade to the console.
   * Expected Output: **The average grade is [calculated value].**
7. Create an object called `*bankAccount*` with properties `*balance*` and `*transactions*` (an array). Implement methods `*deposit*` and `*withdraw*` for the `*bankAccount*` object. The `*deposit*` method should add an amount to the `*balance*`, and the `*withdraw*` method should subtract an amount from the `*balance*`. Call these methods with appropriate values and print the updated balance.
   * Expected Output (after deposits and withdrawals): **The current balance is $[calculated value].**
8. Define an object `*book*` with properties `*title*`, `*author*`, and `*pages*`. Add a method `*info*` to the `*book*` object that returns a string containing information about the book in the format: "Title: [title], Author: [author], Pages: [pages]". Call the `*info*` method and print the book information to the console.
   * Expected Output: **Title: [title], Author: [author], Pages: [pages].**
9. Create an object called `*car*` with properties `*make*`, `*model*`, and `*speed*`. Implement methods `*accelerate*` and `*brake*` for the `*car*` object. The `*accelerate*` method should increase the `*speed*` by a given amount, and the `*brake*` method should decrease the `*speed*` by a given amount. Call these methods with appropriate values and print the updated `*speed*`.
   * Expected Output (after acceleration and braking): **The current speed is [calculated value] km/h.**