Object

1. Create an object representing a car with properties: make, model, and year. Access and log each property.

2. Create a nested object to represent a person with properties: name and address (which should also be an object with street and city).

3. Create an object for a book with properties: title, author, and pages. Modify the number of pages.

4. Create an object to represent a movie, then add a property for the release year.

5. Create an object with at least 5 properties. Use `Object.keys()` to get and log all the keys.

6. Write a function that takes an object and returns an array of keys where the value is a number.

7. Create an object with different data types as values. Use `Object.keys()` to log only the string values.

8. Given an object, log the number of keys it has using `Object.keys()`.

9. Create an object with properties and use `Object.values()` to log the values.

10. Write a function that takes an object and returns the sum of all its numeric values using `Object.values()`.

11. Create an object with mixed types, then filter out and log only the boolean values using `Object.values()`.

12. Create a user object and log all the values in a formatted string.

13. Use `Object.entries()` to log both keys and values of an object in the format "key: value".

14. Write a function that takes an object and returns an array of strings formatted as "key: value".

15. Create an object with properties and log only those entries where the value is a string using `Object.entries()`.

16. Given an object, log the number of key-value pairs it has using `Object.entries()`.

17. Create an object and check if it has a specific property using `Object.hasOwnProperty()`.

18. Write a function that checks if all properties of an object are present and returns true or false using `Object.hasOwnProperty()`.

19. Create an object for a student and check for properties: name and grade.

20. Write a function that takes an object and a property name, returning true if the property exists.

21. Create two objects and merge them into a third using `Object.assign()`.

22. Write a function that takes two objects and returns a new object with properties from both.

23. Create an object representing a user, then use `Object.assign()` to add new properties.

24. Use `Object.assign()` to clone an object and modify a property in the clone.

25. Create an object and freeze it using `Object.freeze()`. Attempt to modify a property and log the result.

26. Write a function that accepts an object and uses `Object.freeze()` to prevent any modifications.

27. Create a frozen object and log an error message when trying to add a new property.

28. Test if an object is frozen using `Object.isFrozen()`.

29. Create an object with at least 10 properties and write a function to count the number of properties of type 'string'.

30. Write a function that takes an object and returns a new object with keys converted to uppercase.

31. Create a function that checks if any value in an object is an array using `Object.values()`.

32. Write a function that takes an object and returns an object with all numeric values squared.

33. Use `Object.keys()`, `Object.values()`, and `Object.entries()` to log an object in a tabular format.

34. Write a function that accepts an object and uses all four methods (`keys`, `values`, `entries`, `hasOwnProperty`) to summarize the object.

35. Create an object and log all keys and their corresponding types using `Object.entries()`.

36. Write a function that combines two objects and logs the conflicting keys and their values.

37. Create a function that accepts an object and a key, returning an error message if the key does not exist.

38. Write a function that ensures a property exists on an object before trying to access it, using `hasOwnProperty()`.

39. Create an object with a method. Ensure the method checks if `this` refers to the correct object.

40. Use `Object.freeze()` on a configuration object and test its immutability.

41. Create a character object for a game and log its properties using `Object.entries()`.

42. Write a function that takes an object and returns an array of keys sorted alphabetically.

43. Create a fruit object and add a method to log a summary of its properties.

44. Use `Object.assign()` to extend a settings object with defaults.

45. Create an employee object, add a method to calculate salary, and log the result.

46. Create a shopping cart object and use methods to add and remove items.

47. Create a settings object and use `Object.freeze()` to lock it down after initial setup.

48. Build a simple library system with books as objects and methods to add and remove books.

49. Create an object representing a recipe with properties for ingredients and instructions, then log it.

50. Write a function that accepts an array of objects and returns a new array of objects with only the specified keys.