**Exercise 1: Accessing Properties**

* ქვემოთ მოცემული ობიექტიდან წაიკითხეთ და დაპრინტეთ **title** და **author** -ის მნიშვნელობები, გამოიყენეთ ორივე ხერხი (წერტილით, ოთხუკთხედი ფრჩხილებით).

const book = {

title: "The Hobbit",

author: "J.R.R. Tolkien",

year: 1937,

};

**Exercise 2: Modifying Properties**

* შეცვალეთ წელი ზემოთ გამოყენებულ ობიექტში რომ გახდეს 1954 და დაპრინტეთ მთლიანი ობიექთი კონსოლში.

**Exercise 3: Adding New Properties**

* დაამატეთ ახალი ფროფერთი - publisher რომლის მნიშვნელობა იქნება - "HarperCollins". დაპრინტეთ მთლიანი ობიექტი

**Exercise 4: Deleting Properties**

* წაშალეთ year ფროფერთი ობიექტიდან. დაპრინტეთ მთლიანი ობიექტი

**2. Object Methods**

**Exercise 5: Defining Methods in Objects**

* დაამატეთ getSummary მეთოდი book ობიექტს რომელიც აბრუნებს ავტორს და სათაურს სტრინგის სახით (მაგ. "The Hobbit by J.R.R. Tolkien"). გამოიძახეთ და დაპრინტეთ შედეგი

const book = {

title: "The Hobbit",

author: "J.R.R. Tolkien",

getSummary() {

// return the summary here

}

};

**3. Computed Property Names**

**Exercise 6: Using Computed Property Names**

* შექმენით student ობიექთი name და age ფროფერთიებით. გამოიყენეთ key ვერიებლი რომ დინამიურად ჩასვათ grade ფროფერთი რომლის მნიშვნელობაც 90 იქნება

let key = "grade";

const student = {

name: "Alice",

age: 20,

};

// dynamically add the grade property here

**4. Object Freezing and Sealing**

**Exercise 7: Object Freezing**

* და- Freeze -ეთ student ობიექტი წინა დავალებაში რაც შექმენით

Object.freeze(student);

student.age = 21; // What happens here?

**Exercise 8: Object Sealing**

* შექმენით car ობიექტი და და -seal ეთ იგი.

const car = {

make: "Toyota",

model: "Camry",

};

// Try adding and deleting properties here

**5. Merging Objects**

**Exercise 9: Merging Objects with Object.assign()**

* დამერჯეთ (გააერთიანეთ) ეს ორი ობიექტი Object.assign() ის დახმარებით.

const person1 = { name: "John", age: 25 };

const person2 = { age: 30, city: "New York" };

**Exercise 10: Merging Objects with Spread Operator**

* იგივე ქენით spread operator(…) - ით.

**6. Cloning Objects**

**Exercise 11: Shallow Copy of an Object**

* Create a shallow copy of the person1 object using the spread operator. Modify a property in the cloned object and observe whether it affects the original object.

const person1 = { name: "John", age: 25, address: { city: "New York" } };

const clonedPerson = { ...person1 };

clonedPerson.name = "Jane"; // Does it affect the original?

**Exercise 12: Deep Copy of an Object**

* შექმენით deep copy person1 ობიექტით.

const person1 = { name: "John", age: 25, address: { city: "New York" } };

**7. Object Iteration Techniques**

**Exercise 13: Using Object.keys()**

* Object.keys()-ის გამოყენებით დაპრინტეთ person1 ის key მნიშვნელობების მასივი

const person1 = { name: "John", age: 25, city: "New York" };

**Exercise 14: Using Object.values()**

* Object.values()ს გამოყენებით დაპრინტეთ ველიუები ზემოთ გამოყენებული ობიექტის.

**Exercise 15: Using Object.entries()**

* Object.entries() -ის გამოყენებით დაპრინტეთ key-value წყვილების მასივი.

const person1 = { name: "John", age: 25, city: "New York" };