Let's break down each task one by one and provide JavaScript solutions for each.

---

1. Shopping Cart Operations

```
const shoppingCart = ['Milk', 'Coffee', 'Tea', 'Honey'];

// Add 'Meat' at the beginning if not already added
if (!shoppingCart.includes('Meat')) {
  shoppingCart.unshift('Meat');
}

// Add 'Sugar' at the end if not already added
if (!shoppingCart.includes('Sugar')) {
  shoppingCart.push('Sugar');
}

// Remove 'Honey' if allergic
const indexOfHoney = shoppingCart.indexOf('Honey');
if (indexOfHoney !== -1) {
  shoppingCart.splice(indexOfHoney, 1);
}

// Modify 'Tea' to 'Green Tea'
const indexOfTea = shoppingCart.indexOf('Tea');
if (indexOfTea !== -1) {
  shoppingCart[indexOfTea] = 'Green Tea';
}

console.log(shoppingCart);
```

2. Operations on Ages Array

```
const ages = [19, 22, 19, 24, 20, 25, 26, 24, 25, 24];
```

```
// Sort the array
ages.sort((a, b) => a - b);

// Find the min and max age
const minAge = ages[0];
const maxAge = ages[ages.length - 1];

// Find the median age
let medianAge;
const middle = Math.floor(ages.length / 2);
if (ages.length % 2 === 0) {
  medianAge = (ages[middle - 1] + ages[middle]) / 2;
} else {
  medianAge = ages[middle];
}

// Find the average age
const averageAge = ages.reduce((sum, age) => sum + age, 0) / ages.length;

// Find the range of the ages
const range = maxAge - minAge;

// Compare (min - average) and (max - average) using abs()
const diffMinAverage = Math.abs(minAge - averageAge);
const diffMaxAverage = Math.abs(maxAge - averageAge);

console.log(`Min Age: ${minAge}`);
console.log(`Max Age: ${maxAge}`);
console.log(`Median Age: ${medianAge}`);
console.log(`Average Age: ${averageAge}`);
console.log(`Range: ${range}`);
```

```
console.log(`Difference (Min - Average): ${diffMinAverage}`);
console.log(`Difference (Max - Average): ${diffMaxAverage}`);
3. Object Extensibility and Sealing
// a) Prevent adding new properties to the student object
const student = { id: 1, firstName: 'John', lastName: 'Doe', age: 21 };
Object.preventExtensions(student);

// b) Check if the student object is extensible
const extensibleStatus = Object.isExtensible(student);

// c) Create a teacher object with 'subject' property
const teacher = { subject: 'Math' };

// d) Seal the teacher object
Object.seal(teacher);

// e) Check if the teacher object is sealed
const sealedStatus = Object.isSealed(teacher);

// f) Print the statuses
console.log(`Student Object Extensible: ${extensibleStatus}`);
console.log(`Teacher Object Sealed: ${sealedStatus}`);
4. Building a Student Management System
const students = [
  { id: 1, firstName: 'Alice', lastName: 'Smith', age: 20, grade: 'A' },
  { id: 2, firstName: 'Bob', lastName: 'Johnson', age: 22, grade: 'B' },
  { id: 3, firstName: 'Charlie', lastName: 'Brown', age: 21, grade: 'A' }
];

// a) Add a Student
function addStudent(newStudent) {
```

```
  students.push(newStudent);
}

// b) Update Student Information
function updateStudent(id, updatedInfo) {
  const student = students.find(student => student.id === id);
  if (student) {
    Object.assign(student, updatedInfo);
  }
}

// c) Delete a Student
function deleteStudent(id) {
  const index = students.findIndex(student => student.id === id);
  if (index !== -1) {
    students.splice(index, 1);
  }
}

// d) List All Students
function listAllStudents() {
  console.log(students);
}

// e) Find Students by Grade
function findStudentsByGrade(grade) {
  return students.filter(student => student.grade === grade);
}

// f) Calculate Average Age
function calculateAverageAge() {
```

```
  const totalAge = students.reduce((sum, student) => sum + student.age, 0);
  return totalAge / students.length;
}

// Example Usage:
addStudent({ id: 4, firstName: 'David', lastName: 'Williams', age: 23, grade: 'C' });
updateStudent(1, { age: 21 });
deleteStudent(2);
listAllStudents();
console.log(findStudentsByGrade('A'));
console.log(calculateAverageAge());
```

5. Using for...in Loop to Iterate Over a Student Object

```
const studentInfo = {
  id: 1,
  firstName: 'John',
  lastName: 'Doe',
  age: 21,
  grade: 'A'
};

// a) Function to display student info using for...in loop
function displayStudentInfo(student) {
  for (let property in student) {
    console.log(`${property}: ${student[property]}`);
  }
}

// Example Usage:
displayStudentInfo(studentInfo);
```

Explanation:

- **Shopping Cart Operations: The program ensures items are added, removed, or modified according to specified conditions.**
- **Ages Array: Operations on sorting, finding min/max, calculating the median, average, range, and differences between min/max and average are done step-by-step.**
- **Object Extensibility and Sealing: The program demonstrates how to prevent additions to an object (prevent extensions) and seal an object to prevent deletions or additions of properties.**
- **Student Management System: This part shows how to create functions for adding, updating, deleting, and listing students, along with calculating average age and filtering by grade.**
- **for...in loop: The loop iterates over the properties of a student object and prints each property and its value.**