1. Create a Student class with name, rollNo, and marks as attributes. Write a method to calculate the grade based on the marks and display the grade. Create multiple instances of the Student class and print their grades.

2. Create a program that keeps track of the number of objects created and displays the count in a function called display().

3. Create an Employee class with members empNo, name, department, and salary. Use a one-dimensional array of Employee objects (size 10) to read the data for 5 employees from command line arguments. Display the data and determine which employee has the highest salary.(you can also used any collection implemented classes (arraylist,set)

4. Create a class Store with attributes StoreID, StoreName, ProductList, and Revenue. Implement methods to add a product to the list, calculate total revenue, and display store information.

5. Create a class Book with data members Title, Author, Cost, and noOfBooks. Implement a function to display information about the book. Also:
6. Create instances of the Book class and initialize the data members.create a List<Book> books and store all the Books in books list
7. Accept a title and noOfBooks from the user. Check if the book exists and if sufficient copies are available. If so, calculate and display the total cost.

8. Create a class Customer with name, address, and phone as attributes. Implement polymorphism by overriding methods in a subclass VIPCustomer to apply discounts for VIP customers.

9. Create class Movie with below attributes: movieName - String company - String genre - String budget - int
   Create class Solution and implement static method "getMovieByGenre" in the Solution class. This method will take array of Movie objects and a searchGenre String as parameters. And will return another array of Movie objects where the searchGenre

String matches with the original array of Movie object's genre attribute (case insensitive search).
Write necessary getters and setters.

Before calling "getMovieByGenre" method in the main method, read values for four Movie objects referring the attributes in above sequence along with a String searchGenre. Then call the "getMovieByGenre" method and write logic in main method to print "High Budget Movie",if the movie budget attribute is greater than 80000000 else print "Low Budget Movie".

10.Create class Player with below attributes: id - int name - String iccRank - int matchesPlayed - int runsScored - int

Create class Solution and implement static method "findAverageOfRuns" in the Solution class. This method will take array of Player objects and a target int as parameters. And will return another double array where the target int is lesser than or equal to the original array of Player object's matchesPlayed attribute and contains the average run scored by each player satisfying above condition.
Write necessary getters and setters.
Before calling "findAverageOfRuns" method in the main method, read values for four Player objects referring the attributes in above sequence along with a int target. Then call the "findAverageOfRuns" method and write logic in main method to print "Grade A",if the calculated averageRun value is 80 to 100. Else if the averageRun value is between 50 and 79 then print "Grade B". Else print "Grade C".

11.Java Program to Remove All Occurrences of an Element from an array.

JAVASCRIPT

Write a JavaScript program to list the properties of a JavaScript object.
Sample object:
var student = {
name : "David Rayy",
sclass : "VI",
rollno : 12 };
Sample Output: name,sclass,rollno

2.Write a JavaScript program to display the reading status (i.e. display book name, author name and reading status) of the following books.

```
var library = [
  {
     author: 'Bill Gates',
     title: 'The Road Ahead',
     readingStatus: true
  },
  {
     author: 'Steve Jobs',
     title: 'Walter Isaacson',
     readingStatus: true
  },
  {
     author: 'Suzanne Collins',
     title:  'Mockingjay: The Final Book of The Hunger Games',
     readingStatus: false
  }];
```

3.Write a JavaScript program to compare two objects to determine if the first contains equivalent property values to the second one.

- Use Object.keys() to get all the keys of the second object.
- Use Array.prototype.every(), Object.prototype.hasOwnProperty() and strict comparison to determine if all keys exist in the first object and have the same values.

  4.Write a JavaScript program to filter out the specified values from a specified array

5. JavaScript code that searches an object from an object array based on an **id** and displays the result in a <div>:

html
CopyEdit

How it works:

1. User enters an **ID** in the input box.
2. When the button is clicked, searchObject() is called.
3. .find() searches the array for an object with a matching id.
4. The result is shown inside the div.