PRACTICAL REVISION SET 1- Object-ify the arrays in PRACTICAL 6B Q7

A PE teacher wants to build a JavaScript program to manage a list of students, with their names, genders, and heights. She has an old program as listed in ANNEX 1, and she wants to update it using Object Oriented Programming (OOP).

QUESTION 1

Based on the information in ANNEX 1, convert the arrays in Annex 1 into a single array of student object literals. Name the array variable fop1A99.

ANNEX 1

```
const readline = require('readline-sync');
const studentName = ["Tommy", "Samuel", "Joy", "Ali", "Lee Lee", "Anne", "David"];
const studentGender = ["M","M","F","M","F","F","M"];
const studentHeight = [174,170,160,165,165,151,140];
```

The fop1A99 array holds student object literals, with the following properties: name, gender, height

Write a function **createObjectsFromArrays** to take in the 3 arrays (studentName, studentGender, studentHeight) previously declared, as input parameters. The function should also take in the array variable fop1A99, which should be updated as an array, with the new object literals.

Test it by printing the contents of the array using console.log().

Sample array created:

```
[
  { name: 'Tommy', gender: 'M', height: 174 },
  { name: 'Samuel', gender: 'M', height: 170 },
  { name: 'Joy', gender: 'F', height: 160 },
  { name: 'Ali', gender: 'M', height: 165 },
  { name: 'Lee Lee', gender: 'F', height: 165 },
  { name: 'Anne', gender: 'F', height: 151 },
  { name: 'David', gender: 'M', height: 140 }
]
```

Question 2:

Write a function **displayNames** that takes in an array of student object literals as parameter. It displays all the student names from the array using a for loop. Test the function by calling it.

Sample output:

```
Names of all students:
Student 1: Tommy
Student 2: Samuel
Student 3: Joy
Student 4: Ali
Student 5: Lee Lee
Student 6: Anne
Student 7: David
```

Question 3:

Create a function **addStudent** that:

- Takes the array of student object literals as parameter,
- prompts user input for name, gender and height of a new record,
- adds the new record into the array, and
- does not return any value.

Invoke the **addStudent** function and enter new student record: 'Pauline', 'F', 158.

Thereafter, invoke the **displayNames** function created in Question 2 to display all student names and check that the new student has been added.

```
Enter name: Pauline
Enter gender: F
Enter height (in cm): 158

Names of all students after adding:
Student 1: Tommy
Student 2: Samuel
Student 3: Joy
Student 4: Ali
Student 5: Lee Lee
Student 6: Anne
Student 7: David
Student 8: Pauline
```

Question 4:

Create a function **createTeams** that:

- Takes in 3 parameters – all students array, girls array, boys array,
- does not return any value, and
- Creates the 2 teams based on the genders of each student object.

Call the **displayNames** function from Question 2 with the girls and boys arrays to check that you have inserted the right students in each team.

Sample output:

```
Girls Team:
Student 1: Joy
Student 2: Lee Lee
Student 3: Anne
Student 4: Pauline

Boys Team:
Student 1: Tommy
Student 2: Samuel
Student 3: Ali
Student 4: David
```

Question 5:

Create a function **computeAverageHeight** that:

1. Takes in an array of student object literals as a parameter, and
2. Computes and returns the average height of all students in that array.

```
Average height of all students: 160cm
Average height of female students: 158cm
Average height of male students: 162cm
```