**ASSIGNMENT 1**

***Lesson 1*** : Input an integer number from keyboard (named n) and

* Print to the screen odd composite numbers < n. (*Composite number is natural numbers greater than 1 and must be divisible by a number > 1 and itself.)*
* Print out n second prime numbers.

- In the natural numbers <= n count how many numbers there are:

+ Divide by 3.

+ Divide 4 with the remainder 1.

+ The end digit is “6”.

* Print out the successful development of prime numbers from small to large.

*Example:*  9 --> 3.3 12 --> 2.2.3

***Lesson 2*** : Given an array of natural numbers

- Print out the screen all prime numbers of this array.

***-*** Count how many prime numbers there are in the above array, how many composite numbers?

- Arranges this array in descending order that items are square numbers.

***Lesson 3:*** Give a students list with the fullname of students.

- Count how many Students that first name is "Yen".

***-*** Count how many of you have the "Van" padding.

***-*** Count how many students that their lastnames starting with the letter "N".

***Lesson 4:*** We need to manage a football team, includes player and coach.

* Player: with the informations: code, name, address, shirt number , position, salary.
* Coach: with the informations: code, name, address, position,salary, years of experience.

1. Build the class as the above requirement (using the inheritance).

2. Build an interface, named IManger to design some methods:

* Input a list of players.
* Input a list of coaches.
* Show list of players.
* Show list of coaches.
* Update the information of players 🡪 void changePlayer(int playercode, int option, int shirtnumber, double salary);

+ If option = 0, change the shirt number of player

+ If option !=0, change the salary.

* Count the coaches that have years of experience >=3.
* Sum of the salary of the players that are the striker.
* Who have the max salary? 🡪 void showMaxLuong();
* Sort the list of players by ascending shirt number.
* Sort descending salaries of experienced coaches = 3.

3. Implement the above interface.

4. Build the main class:

* Show the menu.
* Depending on the option, call the processing functions in section 3 to display the results.

5. Build the class **CheckData**:

* Check null for String properties.
* Check number for number properties.