1. Create a public functional interface Validate with  a method :

**public boolean validateName(String name);**

Create a **public class ValidateUtility** with the below methods :

**public static Validate validateEmployeeName()** – The lambda expression  for the validateName  method must return true if the name is valid and return false if the name is invalid.

In this case, the name is valid if it contains alphabets and space and it should contain minimum 5 characters and maximum 20 characters.

**public static Validate validateProductName()** – The lambda expression  for the validateName  method must return true if the name is valid and return false if the name is invalid.

In this case, the name is valid if the first character is an alphabet followed by 5 digits.

Write the main method in ValidateUtility  class.

             -  Get the value for employee name and product name.

             -  Invoke the validateEmployeeName method

             -  Capture the object of Validate returned by the static method.

             -  Invoke the validateName method for the employee name received as input  from the user.

             -  Display the result as shown in sample output.

             -  Next, invoke the validateProductName method

             -  Capture the object of Validate returned by the static method.

             -  Invoke the validateName method for the product name received as input  from the user.

             -  Display the result as shown in sample output.

Note  :  Implement all the static methods mentioned above using lambda expression.  Don’t create object for the interface using new keyword.

**Sample Input 1 :**

Pinky Rose

A8546

**Sample Output 1 :**

Employee name is valid

Product name is invalid

**Sample Input 2 :**

Rahul@123

X82456

**Sample Output 1 :**

Employee name is invalid

Product name is valid

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

2. Sharpx private company is conducting a conference, for which the participants need to pre-register. After completing the registration they will get an ID. By using the ID they can enter the conference hall. So help Sharpx private company to validate the ID issued to the participants.

**Requirements:**

* The ID should have exactly ten characters.
* The format of the ID is "**SPC00105PM**".
* The first 3 characters of the ID should be "**SPC**", which determines the company name.
* The next 3 characters should be digits, it determines the seat number.
* The last 4 characters should determine the time of the conference.
* Of the last four characters, the first 2 digits indicate time, it should be greater than 0 and less than or equal to 12.
* The next two characters are time zone. It should be either "**AM**" or "**PM**".
* If the input does not match with the above condition, print "**<ID> is an Invalid** **ID**" and terminate the program.

**Note:**

* In the Sample Input / Output provided, the highlighted text in bold corresponds to the input given by the user, and the rest of the text represents the output.
* Ensure to follow the object-oriented specifications provided in the question description.
* Ensure to provide the names for classes, attributes, and methods as specified in the question description.
* Adhere to the code template, if provided.

**Please do not use System.exit(0) to terminate the program**

**Sample Input/Output 1:**

Enter your name

**Aravind**

Enter your ID

**SPC01004PM**

Hi Aravind your seat number is 10 and the event starts at 4PM

**Sample Input/Output 2:**

Enter your name

**Snekha**

Enter your ID

**SPA01004PM**

SPA01004PM is an Invalid ID

**Sample Input/Output 3:**

Enter your name

**Jin**

Enter your ID

**SPC01004HM**

SPC01004HM is an Invalid ID

3. write a demo program to check Supplier interface.