**Question**

Design a class Exchange to accept a sentence and reverse each word in the sentence , with single letter word remaining unchanged. The words in the input sentence are separated by a single blank space and terminated if full stop.

Example : Input – It is a warm day .

Output – tI si a mraw yad

Some of the data members and member functions are given below :

Classname : Exchange

Data members:

sent – stores the sentence

rev – to store new sentence

size – stores the length of the sentence .

Member functions :

Exchange() – default constructor

void readsentence() – to accept the sentence

void exfirstlast():extract each word and interchange the first and last alphabet of the word and form a new sentence rev using the changed words

void display():display the original sentence along with the new changed sentence.

Specify the class Exchange giving details of the constructor(), void readsentence() , void exfirstlast() and void display(). Define the main() function to create an object and call the functions accordingly to enable the task.

**Algorithm**

1. Start

2. Define a class `Exchange` with three instance variables:

- `sent` of type `String` to store the input sentence.

- `rev` of type `String` to store the modified sentence.

- `size` of type `int` to store the length of the sentence.

3. Define a default constructor for the class `Exchange`:

- Initialize `sent` with an empty string.

- Initialize `rev` with an empty string.

- Initialize `size` with 0.

4. Define a method `readsentence()` to accept the sentence from the user:

- Create a `Scanner` object to read input from the user.

- Prompt the user to enter a sentence terminated by a full stop.

- Read the input sentence and store it in `sent`.

- Set `size` to the length of `sent`.

5. Define a method `exfirstlast()` to reverse each word in the sentence:

- Split the sentence `sent` into words using a space as the delimiter.

- Initialize a `StringBuilder` object `newSentence` to build the modified sentence.

- Iterate over each word in the split words:

- If the word ends with a period, remove the period.

- If the word length is greater than 1, reverse the word and append it to `newSentence`.

- Otherwise, append the word as is to `newSentence`.

- Append a space after each word.

- Trim the `newSentence` to remove any trailing spaces and store it in `rev`.

- If the original sentence `sent` ends with a period, append a period to `rev`.

6. Define a method `display()` to display the original and modified sentences:

- Print the original sentence stored in `sent`.

- Print the modified sentence stored in `rev`.

7. In the main method:

- Create an object of the class `Exchange`.

- Call the `readsentence()` method on the object to read the input sentence.

- Call the `exfirstlast()` method on the object to reverse each word in the sentence.

- Call the `display()` method on the object to print the original and modified sentences.

8. End

**Variable Description Table**

|  |  |  |
| --- | --- | --- |
| **Variable Name** | **Data Type** | **Description** |
| sent | String | Stores the input sentence provided by the user. |
| rev | String | Stores the modified sentence after reversing each word. |
| size | int | Stores the length of the input sentence. |
| scanner | Scanner | Used to read input from the user. |
| words | String[] | An array to store the words of the input sentence. |
| newSentence | StringBuilder | Used to build the modified sentence. |
| word | String | A temporary variable to hold each word during iteration. |
| binaryNumber | int | Stores the binary number input by the user (not used in Exchange class). |