Aim: Perform Chunking for the given text input

Objective: To study chunking for a given text.

Theory:

Chunking is a process of extracting phrases from unstructured text, which means analyzing a sentence to identify the constituents(Noun Groups, Verbs, verb groups, etc.) However, it does not specify their internal structure, nor their role in the main sentence. Chunking can break down sentences into phrases that are more useful than single words and provide meaningful outcomes. When extracting information from text, such as places and person names, Chunking is critical. (extraction of entities)

Types:

- Chunking Up

We don't go into great detail here; instead, we're content with a high-level overview. It only serves to provide us with a quick overview of the facts.

- Chunking Down

Unlike the previous method of Chunking, chunking down allows us to obtain more detailed data. Consider "chunking up" if you only need an insight; otherwise, "chunking down" is preferable.

Program:

```
import nltk

text = "The teens wondered what was kept in the red shed on the far edge of the school
grounds."

words = nltk.word_tokenize(text)

pos_tags = nltk.pos_tag(words)

 grammar = r""" NP: {

?*+}"""

chunk_parser = nltk.RegexpParser(grammar)

tree = chunk_parser.parse(pos_tags)

for subtree in tree.subtrees():
```

```
if subtree.label() == 'NP':
        print(' '.join(word for word, tag in subtree.leaves()))
```

Output:

The teens

edge

the school grounds

Conclusion: Chunking is a vital aspect of natural language processing, helping extract meaningful linguistic units from text, including noun phrases and verb phrases. It aids in syntactic parsing, grammar analysis, information extraction, feature extraction for text classification and sentiment analysis, and text summarization. Chunking enhances the efficiency and depth of NLP applications by breaking text into semantically meaningful chunks, improving language understanding and automated processing.