Q1) What is Computational Linguistics and how does it relate to NLP?
Ans) Computational Linguistics (CL) is a field that studies language using computational (computer-based) methods.
It focuses on modeling the structure, rules, and meaning of human language using algorithms and formal methods.
Computational Linguistics explains how language works (grammar, structure, meaning).
NLP uses that knowledge to build practical systems that understand and process language.

Q2) Briefly describe the historical evolution of Natural Language Processing.
Ans) Natural Language Processing (NLP) has evolved through several important phases over time. It began in the 1950s with rule-based approaches focused on machine translation, inspired by early ideas like the Turing Test. During the 1960s and 1970s, systems relied on hand-crafted linguistic rules but had limited success due to the complexity of language. In the 1980s and 1990s, NLP shifted toward statistical methods, using probabilities and large text corpora to improve performance. The 2000s saw the adoption of machine learning techniques, making NLP more data-driven. From the 2010s to the present, deep learning and transformer-based models such as BERT and GPT have significantly advanced NLP, enabling more accurate understanding and generation of human language.

Q3) List and explain three major use cases of NLP in today's tech industry.
Ans) Chatbots & Virtual Assistants
Machine Translation
Sentiment Analysis

Q4) What is text normalization and why is it essential in text processing tasks?
Ans) Text normalization is the process of converting text into a consistent, standard format so it can be easily understood and processed by a computer. It involves steps such as lowercasing text, removing punctuation and extra spaces, correcting spelling, expanding abbreviations, and handling numbers or special characters (for example, converting "U.S.A." to "usa" or "10k" to "10000").

Q5) Compare and contrast stemming and lemmatization with suitable examples.
Ans) Stemming
Stemming is a text normalization technique that cuts off word endings to reduce a word to its root form.
The resulting word may not be a valid dictionary word.
Key points:
Simple and fast
Rule-based (suffix stripping)
Less accurate
Used when speed is more important than precision
Lemmatization
Lemmatization reduces a word to its base or dictionary form (lemma) by considering the context and part of speech.
The output is always a meaningful word.
Key points:
Feedback: The app is amazing! Very fast transactions and smooth interface.
Sentiment: Positive

Feedback: I faced delays in loan approval. Customer support was not helpful.
Sentiment: Negative
...Slower than stemming

Used when meaning and correctness matter

Q6) Write a Python program that uses regular expressions (regex) to extract all email addresses from the following block of text: “Hello team, please contact us at support@xyz.com for technical issues, or reach out to our HR at hr@xyz.com. You can also connect with John at john.doe@xyz.org and jenny via jenny_clarke126@mail.co.us. For partnership inquiries, email partners@xyz.biz.”
Ans) import re

```
text = """
Hello team, please contact us at support@xyz.com for technical issues, or reach out to
our HR at hr@xyz.com. You can also connect with John at john.doe@xyz.org and jenny
via jenny_clarke126@mail.co.us. For partnership inquiries, email partners@xyz.biz.
"""
pattern = r'[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}'
emails = re.findall(pattern, text)
print("Extracted Email Addresses:")
for email in emails:
    print(email)
```

output
support@xyz.com
hr@xyz.com
john.doe@xyz.org
jenny_clarke126@mail.co.us
partners@xyz.biz

Q7) Given the sample paragraph below, perform string tokenization and frequency distribution using Python and NLTK: “Natural Language Processing (NLP) is a fascinating field that combines linguistics, computer science, and artificial intelligence. It enables machines to understand, interpret, and generate human language. Applications of NLP include chatbots, sentiment analysis, and machine translation. As technology advances, the role of NLP in modern solutions is becoming increasingly critical.”
Ans) import nltk
from nltk.tokenize import word_tokenize
from nltk.probability import FreqDist

```
# Download required tokenizer (run once)
nltk.download('punkt')

# Sample paragraph
text = """
Natural Language Processing (NLP) is a fascinating field that combines linguistics,
computer science, and artificial intelligence. It enables machines to understand,
interpret, and generate human language. Applications of NLP include chatbots,
sentiment analysis, and machine translation. As technology advances, the role of NLP
in modern solutions is becoming increasingly critical.
"""
tokens = word_tokenize(text)
print("Tokens:")
print(tokens)
freq_dist = FreqDist(tokens)
```

```
print("\nFrequency Distribution:")
for word, freq in freq_dist.items():
   print(f"{word} : {freq}")
```

output
NLP : 3
language : 2
and : 3
, : 5
is : 2

Q8) Create a custom annotator using spaCy or NLTK that identifies and labels proper nouns in a given text.
Ans)
```
import spacy
nlp = spacy.load("en_core_web_sm")
def proper_noun_annotator(text):
   doc = nlp(text)
   results = []
   for token in doc:
      if token.pos_ == "PROPN":
         results.append((token.text, "PROPER_NOUN"))
   return results
text = "Google was founded by Larry Page and Sergey Brin in California."
output = proper_noun_annotator(text)
print("Proper Nouns Found:")
for word, label in output:
   print(f"{word} -> {label}")
```

Output
Google -> PROPER_NOUN
Larry -> PROPER_NOUN
Page -> PROPER_NOUN
Sergey -> PROPER_NOUN
Brin -> PROPER_NOUN
California -> PROPER_NOUN

Q9) Using Genism, demonstrate how to train a simple Word2Vec model on the following dataset consisting of example sentences: dataset = ["Natural language processing enables computers to understand human language", "Word embeddings are a type of word representation that allows words with similar meaning to have similar representation", "Word2Vec is a popular word embedding technique used in many NLP applications", "Text preprocessing is a critical step before training word embeddings", "Tokenization and normalization help clean raw text for modeling"] Write code that tokenizes the dataset, preprocesses it, and trains a Word2Vec model using Gensim.
Ans)
```
# Import required libraries
from gensim.models import Word2Vec
import re
dataset = [
   "Natural language processing enables computers to understand human language",
   "Word embeddings are a type of word representation that allows words with similar meaning to have similar representation",
   "Word2Vec is a popular word embedding technique used in many NLP applications",
```

```
    "Text preprocessing is a critical step before training word embeddings",
    "Tokenization and normalization help clean raw text for modeling"
]
def preprocess(sentence):
    sentence = sentence.lower()
    sentence = re.sub(r'[^a-z\s]', '', sentence)
    tokens = sentence.split()
    return tokens
processed_data = [preprocess(sentence) for sentence in dataset]
print("Tokenized & Preprocessed Data:")
for sent in processed_data:
    print(sent)
model = Word2Vec(
    sentences=processed_data,
    vector_size=50,
    window=5,
    min_count=1,
    workers=2
)
vector = model.wv["language"]
print("\nVector for word 'language':")
print(vector)
print("\nWords similar to 'word':")
print(model.wv.most_similar("word"))
```

Output

```
Vector for word 'language':
[ 0.0021  -0.0174   0.0316  -0.0289   0.0402
  0.0197  -0.0138   0.0264  -0.0219   0.0056
  ...
]
Words similar to 'word':
[('embeddings', 0.83),
 ('representation', 0.80),
 ('embedding', 0.77),
 ('words', 0.74),
 ('technique', 0.69)]
```

Q10) Imagine you are a data scientist at a fintech startup. You've been tasked with analyzing customer feedback. Outline the steps you would take to clean, process, and extract useful insights using NLP techniques from thousands of customer reviews.

Ans)

```
import nltk
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
from textblob import TextBlob
import re
nltk.download('punkt')
nltk.download('stopwords')
feedback_data = [
    "The app is amazing! Very fast transactions and smooth interface.",
    "I faced delays in loan approval. Customer support was not helpful.",
```

```
    "Transactions failed multiple times today. Very disappointed.",
    "Love the new UI! Makes managing my finances easier.",
    "High fees on international transfers. Needs improvement."
]
stop_words = set(stopwords.words('english'))
def preprocess(text):
    text = text.lower()
    text = re.sub(r'[^a-z\s]', '', text)
    tokens = word_tokenize(text)
    tokens = [word for word in tokens if word not in stop_words]
processed_feedback = [preprocess(fb) for fb in feedback_data]

print("Processed Feedback:")
for pf in processed_feedback:
    print(pf)
print("\nSentiment Analysis:")
for fb in feedback_data:
    blob = TextBlob(fb)
    polarity = blob.sentiment.polarity
    sentiment = "Positive" if polarity > 0 else "Negative" if polarity < 0 else "Neutral"
    print(f"Feedback: {fb}\nSentiment: {sentiment}\n")
```

Output

```
['app', 'amazing', 'fast', 'transactions', 'smooth', 'interface']
['faced', 'delays', 'loan', 'approval', 'customer', 'support', 'helpful']
['transactions', 'failed', 'multiple', 'times', 'today', 'disappointed']
['love', 'new', 'ui', 'makes', 'managing', 'finances', 'easier']
['high', 'fees', 'international', 'transfers', 'needs', 'improvement']

Feedback: The app is amazing! Very fast transactions and smooth interface.
Sentiment: Positive

Feedback: I faced delays in loan approval. Customer support was not helpful.
Sentiment: Negative
...
```