Q1. Explain the difference between greedy and non-greedy syntax with visual terms and minimal effort to transform patterns. What changes can you make?

- Answer: Greedy matches capture the longest possible text, while non-greedy matches capture the shortest. To make a greedy pattern non-greedy, add a `?` after quantifiers, e.g., `\*?` or `+?`.

Q2. When does greedy vs. non-greedy matter? What if only a greedy match is available?

- Answer: Greediness matters when matching text with multiple occurrences. If only a greedy match is available, it may capture more than desired.

Q3. In a simple string match without replacement, does a non-tagged group matter?

- Answer: In simple matching, non-tagged groups have no practical difference.

Q4. Describe a scenario where using a non-tagged category significantly impacts program outcomes.

- Answer: Non-tagged groups are usually less impactful; tagged groups provide named references.

Q5. How does look-ahead condition not consuming characters affect program results?

- Answer: Look-ahead conditions examine characters without consuming them, useful when you need to match based on context without including the context in the match.

Q6. What's the difference between positive and negative look-ahead in regular expressions?

- Answer: Positive look-ahead (`(?=...)`) asserts that a pattern must follow, while negative look-ahead (`(?!...)`) asserts that a pattern must not follow at a specific position.

Q7. Why is referring to groups by name better than by number in regular expressions?

- Answer: Referring to groups by name makes regex patterns more readable and maintainable, especially in complex expressions.

Q8. Can you identify repeated items within a string using named groups, e.g., "The cow jumped over the moon"?

- Answer: Named groups can identify repeated items in a string, but in this example, without repetition, it may not apply.

Q9. What does the Scanner interface do that re.findall doesn't when parsing a string?

- Answer: The Scanner interface in Python provides more fine-grained control and customization compared to the simpler `re.findall` function.

Q10. Must a Scanner object be named 'scanner'?

- Answer: No, a Scanner object can be named as desired; 'scanner' is just a variable name.