**Q1. Explain the difference between greedy and non-greedy syntax with visual terms in as few words**

**as possible. What is the bare minimum effort required to transform a greedy pattern into a non-greedy**

**one? What characters or characters can you introduce or change?**

**Q2. When exactly does greedy versus non-greedy make a difference?  What if you&#39;re looking for a**

**non-greedy match but the only one available is greedy?**

**Q3. In a simple match of a string, which looks only for one match and does not do any replacement, is**

**the use of a nontagged group likely to make any practical difference?**

**Q4. Describe a scenario in which using a nontagged category would have a significant impact on the**

**program&#39;s outcomes.**

**Q5. Unlike a normal regex pattern, a look-ahead condition does not consume the characters it**

**examines. Describe a situation in which this could make a difference in the results of your**

**programme.**

**Q6. In standard expressions, what is the difference between positive look-ahead and negative look-**

**ahead?**

**Q7. What is the benefit of referring to groups by name rather than by number in a standard**

**expression?**

**Q8. Can you identify repeated items within a target string using named groups, as in &quot;The cow**

**jumped over the moon&quot;?**

**Q9. When parsing a string, what is at least one thing that the Scanner interface does for you that the**

**re.findall feature does not?**

**Q10. Does a scanner object have to be named scanner?**

**SOLUTIONS**

***1. Greedy syntax matches the longest possible string, while non-greedy matches the shortest possible. To transform a greedy pattern into a non-greedy one, append a "?" to the end of the quantifier.***

***2. Greedy versus non-greedy makes a difference when the pattern has multiple possible matches. If a non-greedy match is required, the pattern must be transformed, and if only a greedy match is available, the desired result may not be achievable.***

***3. The use of a non-tagged group will not make any practical difference in a simple match of a string that looks for only one match and does not do any replacement.***

***4. A scenario in which using a non-tagged category could have a significant impact on the program's outcomes is when the pattern needs to be matched repeatedly, as named categories allow for the repeated reference to a single pattern.***

***5. Look-ahead conditions are useful in cases where the match depends on characters that should not be consumed by the regex pattern, such as when matching overlapping patterns.***

***6. Positive look-ahead asserts that the pattern must be followed by a specific sequence, while negative look-ahead asserts that the pattern must not be followed by a specific sequence.***

***7. Referring to groups by name rather than by number makes the regular expression more readable and easier to understand.***

***8. Yes, repeated items within a target string can be identified using named groups.***

***9. The Scanner interface allows for tokenization, which breaks the input string into smaller parts. re.findall, on the other hand, returns a list of all non-overlapping matches.***

***10. No, a Scanner object does not have to be named "scanner". It can be named anything the programmer chooses.***

***Top of Form***