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?

Ans : It means the greedy quantifiers will match their preceding elements as much as possible to return to the biggest match possible. On the other hand, the non-greedy quantifiers will match as little as possible to return the smallest match possible. non-greedy quantifiers are the opposite of greedy ones.

So the difference between the greedy and the non-greedy match is the following: The greedy match will try to match as many repetitions of the quantified pattern as possible. The non-greedy match will try to match as few repetitions of the quantified pattern as possible.

Characters may change: The way they think and feel about a particular person or group of people. For example, they may come to realise that an enemy or opponent is no different from them.

Q2. When exactly does greedy versus non-greedy make a difference?  What if you're looking for a non-greedy match but the only one available is greedy?

Ans : So the difference between the greedy and the non-greedy match is the following: The greedy match will try to match as many repetitions of the quantified pattern as possible. The non-greedy match will try to match as few repetitions of the quantified pattern as possible.

'Greedy' means match longest possible string. 'Lazy' means match shortest possible string.

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's outcomes.

Ans : Taxonomies of Learning Outcomes can provide a structure to help you think through your course goals and objectives, which then provide the basis for everything else in the course, particularly the assessments.

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.

Ans : Lookarounds often cause confusion to the regex apprentice. I believe this confusion promptly disappears if one simple point is firmly grasped. It is that at the end of a lookahead or a lookbehind, the regex engine hasn't moved on the string. You can chain three more lookaheads after the first, and the regex engine still won't move. In fact, that's a useful technique.

* Turn potential weaknesses into learning opportunities. Instead of dwelling on the regret or what you would have done differently, focus on the positive. ...
* Talk about how you'd handle the situation now. It can be helpful to talk about how you'd deal with a similar situation now. ...
* Be honest.

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

Ans : Positive lookahead: (?= «pattern») matches if pattern matches what comes after the current location in the input string. Negative lookahead: (?! «pattern») matches if pattern does not match what comes after the current location in the input string.

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

Ans : Now, with named groups, we can name each match in the regular expression. So instead of referencing matches of the regular expression with numbers (group(1), group(2), etc.), we can reference matches with names, such as group('month'), group('day'), group('year').

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

Ans : Here's one possible use (out of zillions): to transform a target string by replacing each repeated pair of words with just one word.

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?

Ans : A simple text scanner which can parse primitive types and strings using regular expressions. A Scanner breaks its input into tokens using a delimiter .