```
1  from data_class import Data
2  from loadingModule_2 import loadingAnimation
3
4  # method that returns the question-type like 'what', 'when', where' etc...
5  def printAllQuestionType(processed_data):
6      if len(processed_data.get("QT")) == 1:
7          return """ + processed_data.get("QT").__getitem__(0) + """
8      else:
9          result = "mix of "
10         for c in range(len(processed_data.get("QT")) - 1):
11             result += """ + processed_data.get("QT").__getitem__(c) + "', "
12         result += "and '" + processed_data.get("QT").__getitem__(len(processed_data.get("QT")) - 1) + """
13     return result
14
15 # method that returns any possible places that the user mentioned in their original statement ~ improves
   specific contextualization
16 def printPossibleMentions(saved_input, list):
17     result = ""
18     for places in list:
19         if places.lower() in saved_input.lower():
20             result += places + ", "
21     return result
22
23 # introductory instructions and terms/conditions
24 print("QUESTION ANALYZER: \n1.) Enter a question that you might ask a Google Assistance, Alexa,
   Siri, Cortana, etc..."
25     "\n2.) The system will process your response.\n3.) It will give you it's understanding of the
   question by categorizing it using basic Natural Language Processing (NLP) algorithm.\n")
26 print("This is a first step taken to complete a part of 'robotics data interpretation' and will be built
   upon modularly.\n")
27 print("This model is still in BETA; some questions might not be recognizable by the system. More
   updates will be rolling out soon. [Version: 1.5]\n\n")
28 user_input = input("Ask me anything: ").lower()
29 data_instance = Data()
30 rowCategory = None
31
32 # loop helps the user continue asking more questions for system categorization
33 while user_input != "stop":
34     saved_input = user_input
35     processed_data = data_instance.parsedData(data_instance.stemWord(user_input))
36
37     # ensures there are no error, else, redirect the issue
38     if processed_data.get("I") != 'ERROR':
39         loadingAnimation()
40         for row in range(len(data_instance.possibleList)):
41             for column in range(len(data_instance.possibleList[row])):
42                 for a in range(len(processed_data.get("I"))):
43                     if data_instance.possibleList[row][column] == processed_data.get("I").__getitem__(a):
44                         rowCategory = str(row)
45         print(f"\n\n\nI understand that you are trying to ask a question that starts with a {
   printAllQuestionType(processed_data)} and I am supposed to give a(n) ")
46
47         # list of possible type of questions extracted from user input [non-exhaustive]
48         if rowCategory == "0":
49             print("concise and short answer for general information related to places, conditions,
   activities, person.")
50         elif rowCategory == "1":
51             print("personal/health based information regarding diet/lifestyle.")
52         elif rowCategory == "2":
53             print("productivity tailored response for improving your health/work/performance
   efficiency.")
54         elif rowCategory == "3":
55             print("entertainment related answer that is tailored to improving your leisure/mood.")
56         elif rowCategory == "4":
57             print("mathematical answer, which can either be short or informative, for a concept or
   calculation.")
58         elif rowCategory == "5":
59             print("response that improves the your knowledge in crucial details regarding current
   events.")
60         elif rowCategory == "6":
61             print("advice to the you in the best manner possible that will give you better guidance.")
62         if (printPossibleMentions(saved_input, data_instance.specificPlaceList) != ""):
63             print("You mentioned " + printPossibleMentions(saved_input, data_instance.specificPlaceList
```

```
63  ) + "meaning you want my responses to be personalized for that/those place(s)."
64          if (printPossibleMentions(saved_input, data_instance.specificPopCultureList) != ""):
65              print("You also mentioned Iconic References like " + printPossibleMentions(saved_input,
    data_instance.specificPopCultureList) + "therefore my answers should be refined to that/those
    reference(s).")
66          user_input = input("\nIs my understanding right? Type 'Y' for Yes or 'N' for No: ")
67          print("Glad I am doing it right. Data has been noted!") if user_input == "Y" else input("In what
    way should I have interpreted the response: ")
68          print("Thank you for your feedback!\n")
69          user_input = input("\nAsk me anything (or type 'stop' to end): ")
70      else:
71          user_input = input("\nThis question is unrecognizable. Try again or type 'stop': ")
```