#Create a class call VARK.

#(The class named VARK is defined by the code to encapsulate the VARK questionnaire logic.)

class VARK:

    #Define a Init function.

    #(This is a init (short form of initialization) function which stores the dictionaries.

    #(The VARK class is initialized with dictionaries for questions, answers, an answer guide, user answers and scores.)

    def \_\_init\_\_(self):

            #Dictionary of Questions.

            #(This dictionary stores the 16 questions, each with a unique number.)

        self.questions = {

            1: "You are helping someone who wants to go to your airport, town centre or railway station. You would:",

            2: "You are not sure whether a word should be spelled `dependent' or `dependant'. You would:",

            3: "You are planning a holiday for a group. You want some feedback from them about the plan. You would:",

            4: "You are going to cook something as a special treat for your family. You would:",

            5: "A group of tourists want to learn about the parks or nature reserves in your area. You would:",

            6: "You are about to purchase a digital camera or mobile phone. Other than price,what would most influence your decision?:",

            7: "Remember a time when you learned how to do something new. Try to avoid choosing a physical skill, e.g. riding a bike. You learned best by:",

            8: "You have a problem with your knee. You would prefer that the doctor:",

            9: "You want to learn a new programme,skill or game on a computer. You would:",

            10: "I like websites that have:",

            11: "Other than price,what would most influence your decision to buy a new non-fiction book?",

            12: "You are using a book, DVD or website to learn how to take photos with your new digital camera. You would like to have:",

            13: "Do you prefer a trainer or a presenter who uses:",

            14: "You have finished a competition or test and would like some feedback. You would like to have feedback:",

            15: "You are going to choose food at a restaurant or cafe. You would:",

            16: "You have to make an important speech at a conference or special occasion. You would:"

        }

        #Dictionary of Answers.

        #(This dictionary stores the four (4) optional answers for each questions respectively.)

        #(The answers are arranged in keys and values. Keys are represented by letters A, B, C and D while, values are represented by sentences.)  )

        self.answers = {

            1: {"A": "go with her.", "B": "tell her the directions.", "C": "write down the directions.", "D": "draw, or give her a map."},

            2: {"A": "see the words in your mind and choose by the way they look.", "B": "think about how each word sounds and choose one.", "C": "find it in a dictionary.", "D": "write both words on paper and choose one."},

            3: {"A": "describe some of the highlights.", "B": "use a map or website to show them the places.", "C": "give them a copy of the printed itinerary.", "D": "phone, text or email them."},

            4: {"A": "cook something you know without the need for instructions.", "B": "ask friends for suggestions.", "C": "look through the cookbook for ideas from the pictures.", "D": "use a cookbook where you know there is a good recipe."},

            5: {"A": "talk about, or arrange a talk for them about parks or nature reserves.", "B": "show them internet pictures, photographs or picture books.", "C": "take them to a park or nature reserve and walk with them.", "D": "give them a book or pamphlets about the parks or nature reserves."},

            6: {"A": "Trying or testing it.", "B": "Reading the details about its features.", "C": "It is a modern design and looks good.", "D": "The salesperson telling me about its features."},

            7: {"A": "watching a demonstration.", "B": "listening to somebody explaining it and asking questions.", "C": "diagrams and charts - visual clues.", "D": "written instructions - e.g. a manual or textbook."},

            8: {"A": "gave you a web address or something to read about it.", "B": "used a plastic model of a knee to show what was wrong.", "C": "described what was wrong.", "D": "howed you a diagram of what was wrong."},

            9: {"A": "read the written instructions that came with the programme.", "B": "with people who know about the programme.", "C": "use the controls or keyboard.", "D": "follow the diagrams in the book that came with it."},

            10: {"A": "things I can click on, shift or try.", "B": "interesting design and visual features.", "C": "interesting written descriptions, lists and explanations.", "D": "audio channels where I can hear music, radio programmes or interviews."},

            11: {"A": "The way it looks is appealing.", "B": "Quickly reading parts of it.", "C": "A friend talks about it and recommends it.", "D": "It has real-life stories, experiences and examples."},

            12: {"A": "a chance to ask questions and talk about the camera and its features.", "B": "clear written instructions with lists and bullet points about what to do.", "C": "diagrams showing the camera and what each part does.", "D": "many examples of good and poor photos and how to improve them."},

            13: {"A": "demonstrations, models or practical sessions.", "B": "question and answer, talk, group discussion, or guest speakers.", "C": "handouts, books, or readings.", "D": "diagrams, charts or graphs."},

            14: {"A": "using examples from what you have done.", "B": "using a written description of your results.", "C": "from somebody who talks it through with you.", "D": "using graphs showing what you had achieved."},

            15: {"A": "choose something that you have had there before.", "B": "listen to the waiter or ask friends to recommend choices.", "C": "choose from the descriptions in the menu.", "D": "look at what others are eating or look at pictures of each dish."},

            16: {"A": "make diagrams or get graphs to help explain things.", "B": "write a few key words and practice saying your speech over and over.", "C": "write out your speech and learn from reading it over several times.","D": "gather many examples and stories to make the talk real and practical."}

        }

        #Dictionary of Answer Guid.

        #(This dictionary stores the answer guid for each of the 16 question, in which each optional answers (A, B, C and D) are matched with each of the learning preferences (VARK) respectively.

        self.answer\_guide = {

            1:{"A":"K","B":"A","C":"R","D":"V"},

            2:{"A":"V","B":"A","C":"R","D":"K"},

            3:{"A":"K","B":"V","C":"R","D":"A"},

            4:{"A":"K","B":"A","C":"V","D":"R"},

            5:{"A":"A","B":"V","C":"K","D":"R"},

            6:{"A":"K","B":"R","C":"V","D":"A"},

            7:{"A":"K","B":"A","C":"V","D":"R"},

            8:{"A":"R","B":"K","C":"A","D":"V"},

            9:{"A":"R","B":"A","C":"K","D":"V"},

            10:{"A":"K","B":"V","C":"R","D":"A"},

            11:{"A":"V","B":"R","C":"A","D":"K"},

            12:{"A":"A","B":"R","C":"V","D":"K"},

            13:{"A":"K","B":"A","C":"R","D":"V"},

            14:{"A":"K","B":"R","C":"A","D":"V"},

            15:{"A":"K","B":"A","C":"R","D":"V"},

            16:{"A":"V","B":"A","C":"R","D":"K"}

        }

    # Define a unique function call run\_test.

    # (This function sets out the introduction of the output, the output of the question with answers,

    # and taking in users input.)

    def run\_test(self):

        #This is the introduction and the instruction information to the users.

        print("Welcome to the VARK QUESTIONNAIRE LEARNING METHOD!"

        "\nPlease complete the following VARK questionnaire. "

        "Choose the answer which best explains your preference and enter the letter(s) next to it. "

        "\nPlease enter more than one if a single answer does not match your perception. "

        "Leave blank any question that does not apply.")

        #This is an empty dictionary to store the user's answers.

        user\_answers = {}

        #This method prints out a question and its answer options.

        for question\_num, question in self.questions.items():

            print(f"\nQuestion{question\_num}. {question}")

            for answer\_key, answer\_text in self.answers[question\_num].items():

                print(f"   {answer\_key}: {answer\_text}")

            #This method iterates through quuestions, prompts for answers(one or two) and validates input.

            while True:

                user\_input = input("Enter your answer(s) (e.g., A/B/C/D or AB): ").upper()

                if all(ans in "ABCD" for ans in user\_input) and len(user\_input) <= 2:

                    user\_answers[question\_num] = user\_input

                    break

                else:

                    print("Invalid input. Please enter one or two of the letters A, B, C, or D.")

        self.calculate\_score(user\_answers)

    #This method iterates through user answers.

    def calculate\_score(self, user\_answers):

        #This is the dictionary of scores.

        scores = {"V": 0, "A": 0, "R": 0, "K": 0}

        for question\_num, answers in user\_answers.items():

            for answer in answers:

                scores[self.answer\_guide[question\_num][answer]] += 1

        #This is two rows of dashed parallel lines at the end of the 16th question.

        print("-"\*100)

        print("-"\*100)

        #This is output of the users VARK score.

        print("\nYour VARK scores:")

        for category, score in scores.items():

            print(f"{category}: {score}")

        self.interpret\_results(scores)

    #This method interprets the results and determines the preference (V,A,R,K) and updates scores accordingly.

    def interpret\_results(self, scores):

        print("\nInterpretation of calculated results:")

        max\_score = max(scores.values())

        max\_categories = [cat for cat, score in scores.items() if score == max\_score]

       #This prints the preference for one dominant learning preference.

        if len(max\_categories) == 1:

            print(f"You have a dominant learning preference {max\_categories[0]}.")

            print(f"Therefore, your dominant VARK Learning preference is {max\_categories[0]}.")

        #This prints the preferences of multiple (two) dominant learning preference.

        else:

            print(f"You have multiple dominant learning preferences of: {'and '.join(max\_categories)}.")

            print(f"Therefore, your multiple dominant VARK Learning preferences are: {'and '.join(max\_categories)}.")

        #This is some further interpretation based on VARK learning styles/preferences and also the closing remarks of the program.

        print("\nFurther interpretation based on VARK learning styles:")

        print("V (Visual): You prefer learning through images, charts, and other visual aids.")

        print("A (Aural): You prefer learning through listening, discussions, and audio materials.")

        print("R (Reading/Writing): You prefer learning through written text, notes, and reading materials.")

        print("K (Kinesthetic): You prefer learning through hands-on activities, experiments, and real-world examples.")

        print("\nThank you for doing this particular VARK Questions. " \

        "\nBy now you should be able to identify which type of learner you are, "

        "\nand you can unclock your potential in acquiring as much knowledge as possible.")

#This is the end of the program, thus it return back to the function and class  that was called.

if \_\_name\_\_ == "\_\_main\_\_":

    vark\_test = VARK()

    vark\_test.run\_test()