Title Of The Project : Online Quiz System

Student Name: Mansi / Harshita **UID:** 24MCC20025

24MCC20024

Branch: MCA (CCD) **Section/Group:** 1A

Semester: 1st Sem **Date of Performance:** 24/10/2024

Subject Name: Python Programming **Subject Code:** 24CAH_606

Aim/Overview of the practical:

The Aim of this project is to create a simple, interactive quiz in Python that helps people learn key programming concepts. The quiz will:

- Check Understanding: Test how well users know basic Python ideas, like how to use operators, work with different data types, and understand Python's syntax.
- Build Coding Skills: Give users a chance to practice coding by working on a quiz that asks for their input, checks their answers, and changes the questions each time for variety.
- Support Learning with Feedback: After each question, users will know right away if they got it correct, helping them learn from both their successes and mistakes.
- Boost Problem-Solving: Encourage users to think carefully when choosing their answers, and help them understand why an answer was right or wrong.

Task to be done:

To prepare and complete this project of creating an interactive Python quiz system, the following tasks need to be done:

1. **Define the Quiz Content**

- **Task**: Prepare a set of Python-related questions. Focus on key concepts such as data types, operators, control flow, syntax, and functions.
- **Output**: A list of questions with multiple-choice answers and correct solutions.

2. **Design the Quiz Structure**

- **Task**: Decide how the quiz will flow. Will it be timed? How many questions will users answer? Should the questions be randomly shuffled each time?
- **Output**: A clear plan for the quiz structure (e.g., 5 random questions per session, multiple-choice options, etc.).

3. **Code the Quiz Logic**

- **Task**: Write Python code that:
  - Displays the questions and answer options.
  - Takes user input.
  - Validates answers.
  - Tracks the user's score.
  - Provides instant feedback after each question.

  Output: A functional Python script that runs the quiz interactively.

4. **Implement Input Validation**

- **Task**: Ensure that the quiz accepts only valid inputs (e.g., numbers corresponding to the options) and handles errors smoothly (e.g., warning for invalid inputs).
- **Output**: Robust error-handling code that ensures smooth user interaction.

5. **Add Feedback Mechanism**

- **Task**: Implement feedback that tells users whether their answer was correct or wrong, and provide the correct answer when they make a mistake.
- **Output**: Code that gives clear feedback after each question.

6. **Randomize Questions**

- **Task**: Shuffle the order of questions for each quiz attempt to create a dynamic experience for users.
- **Output**: Code that randomizes questions for each session.

7. Test and Debug

- **Task**: Thoroughly test the quiz to identify and fix any bugs (e.g., incorrect answer handling, input errors).
- **Output**: A bug-free and smoothly running quiz program.

8. User Interface (Optional)

- **Task**: Design a simple and user-friendly interface for displaying questions and collecting answers, which can be text-based (command line) or graphical (using a library like Tkinter, if desired).
- **Output**: A clean interface where users can interact with the quiz easily.

9. Project Documentation

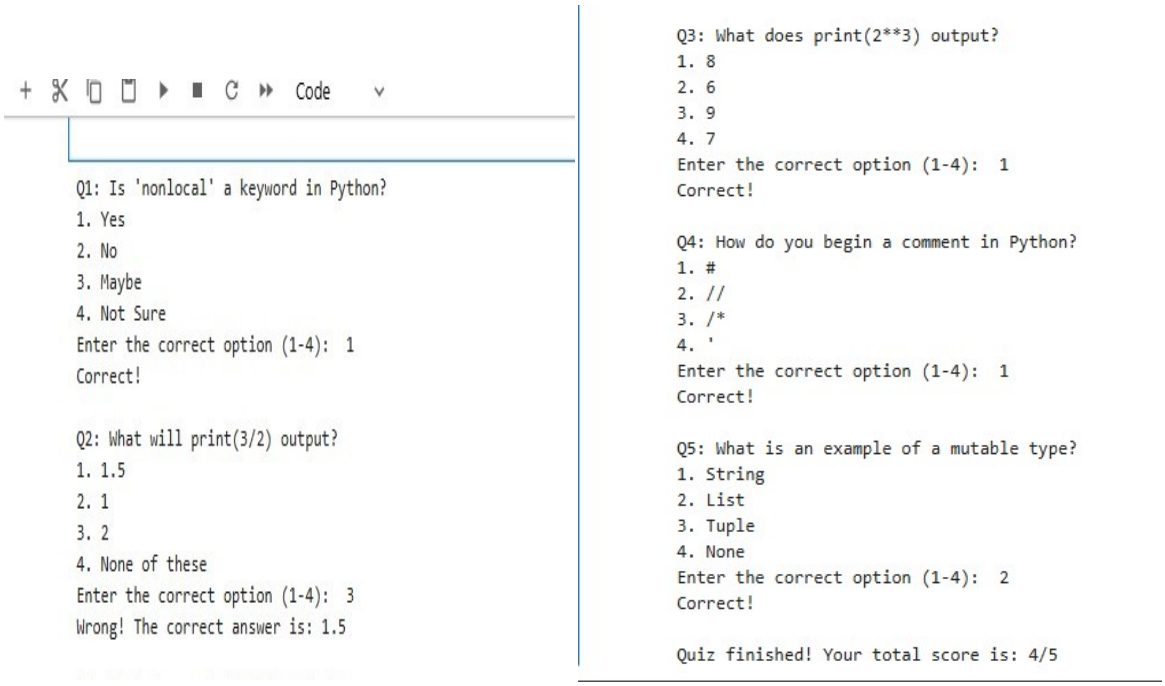
- **Task**: Write clear instructions on how to use the quiz and explain the code for future reference or further improvement.
- **Output**: A document explaining the project, how to run it, and the key components of the code.

Code for experiment/practical:

```
import random

# Python Quiz Questions List
questions = [
    {"question": "What does print(2**3) output?", "options": ["8", "6", "9", "7"], "answer": 8},
    {"question": "What is an example of a mutable type?", "options": ["String", "List", "Tuple", "None"], "answer": "List"},
    {"question": "How do you begin a comment in Python?", "options": ["#", "//", "/*", """], "answer": "#"},
    {"question": "Is 'nonlocal' a keyword in Python?", "options": ["Yes", "No", "Maybe", "Not Sure"], "answer": "Yes"},
```

```
    {"question": "What will print(3/2) output?", "options": ["1.5", "1", "2", "None of these"],
"answer": "1.5"}
]

# Function to run the quiz
def run_quiz():
    score = 0  # Initialize score

    random.shuffle(questions)  # Shuffle questions to create a dynamic quiz session

    # Loop through the first 5 questions
    for i, q in enumerate(questions[:5]):
        print(f"Q{i+1}: {q['question']}")

        # Display options
        for idx, option in enumerate(q['options']):
            print(f"{idx+1}. {option}")

        # Take user's input and validate it
        while True:
            answer = input("Enter the correct option (1-4): ")
            if answer.isdigit() and 1 <= int(answer) <= len(q['options']):
                break  # A valid answer, exit the loop
            else:
                print("Invalid input! Enter a number from 1 to", len(q['options']))

        # Check the user's answer
        if q['options'][int(answer)-1] == str(q['answer']) or q['options'][int(answer)-1] ==
q['answer']:
            score += 1
            print("Correct!\n")
        else:
            print(f"Wrong! The correct answer is: {q['answer']}\n")

    print(f"Quiz finished! Your total score is: {score}/{len(questions[:5])}")

# Run the quiz
run_quiz()
```

Result/Output/Writing Summary:



```
Code

Q1: Is 'nonlocal' a keyword in Python?
1. Yes
2. No
3. Maybe
4. Not Sure
Enter the correct option (1-4):  1
Correct!

Q2: What will print(3/2) output?
1. 1.5
2. 1
3. 2
4. None of these
Enter the correct option (1-4):  3
Wrong! The correct answer is: 1.5

Q3: What does print(2**3) output?
1. 8
2. 6
3. 9
4. 7
Enter the correct option (1-4):  1
Correct!

Q4: How do you begin a comment in Python?
1. #
2. //
3. /*
4. '
Enter the correct option (1-4):  1
Correct!

Q5: What is an example of a mutable type?
1. String
2. List
3. Tuple
4. None
Enter the correct option (1-4):  2
Correct!

Quiz finished! Your total score is: 4/5
```

Learning outcomes (What I have learnt):

I. **Understanding of Python Basics :** By creating and answering quiz questions, you will deepen your knowledge of fundamental Python concepts like data types, operators, syntax, and keywords.

II. **Hands-on Coding Experience :** You'll practice writing Python code, including taking user input, validating responses, and implementing basic control flow (loops, if-statements), which strengthens your programming skills.

III. **Problem-Solving Skills :** Working on the quiz logic, such as handling errors and checking answers, will enhance your ability to think critically and solve coding problems effectively.

IV. **User Interaction and Input Validation :** You'll learn how to interact with users through input and output, and how to ensure the program works smoothly by validating user input and handling mistakes.

V. **Code Organization and Structure :** By building this project, you'll understand how to organize your code logically, making it easier to read, modify, and expand later.

VI. **Debugging and Testing :** Testing and fixing errors in your program will teach you valuable debugging skills and help you develop a mindset of writing error-free code.