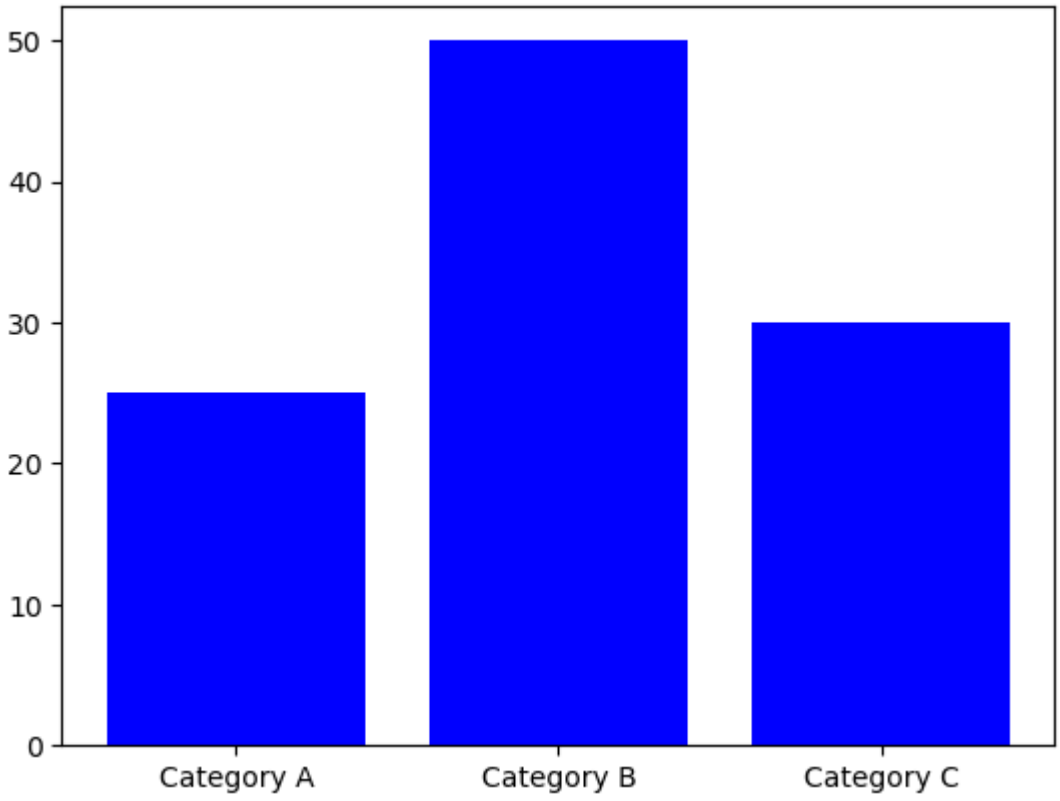
In [1]:
```
pip install matplotlib
```

```
Requirement already satisfied: matplotlib in c:\users\tejasri moyyi\anaconda3\lib\sit
e-packages (3.5.2)Note: you may need to restart the kernel to use updated packages.

Requirement already satisfied: cycler>=0.10 in c:\users\tejasri moyyi\anaconda3\lib\s
ite-packages (from matplotlib) (0.11.0)
Requirement already satisfied: pillow>=6.2.0 in c:\users\tejasri moyyi\anaconda3\lib
\site-packages (from matplotlib) (9.2.0)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\tejasri moyyi\anacond
a3\lib\site-packages (from matplotlib) (2.8.2)
Requirement already satisfied: pyparsing>=2.2.1 in c:\users\tejasri moyyi\anaconda3\l
ib\site-packages (from matplotlib) (3.0.9)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\tejasri moyyi\anaconda3
\lib\site-packages (from matplotlib) (4.25.0)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\tejasri moyyi\anaconda3
\lib\site-packages (from matplotlib) (1.4.2)
Requirement already satisfied: packaging>=20.0 in c:\users\tejasri moyyi\anaconda3\li
b\site-packages (from matplotlib) (21.3)
Requirement already satisfied: numpy>=1.17 in c:\users\tejasri moyyi\anaconda3\lib\si
te-packages (from matplotlib) (1.21.5)
Requirement already satisfied: six>=1.5 in c:\users\tejasri moyyi\anaconda3\lib\site-
packages (from python-dateutil>=2.7->matplotlib) (1.16.0)
```

In [2]:
```
import matplotlib.pyplot as plt
import numpy as np
```

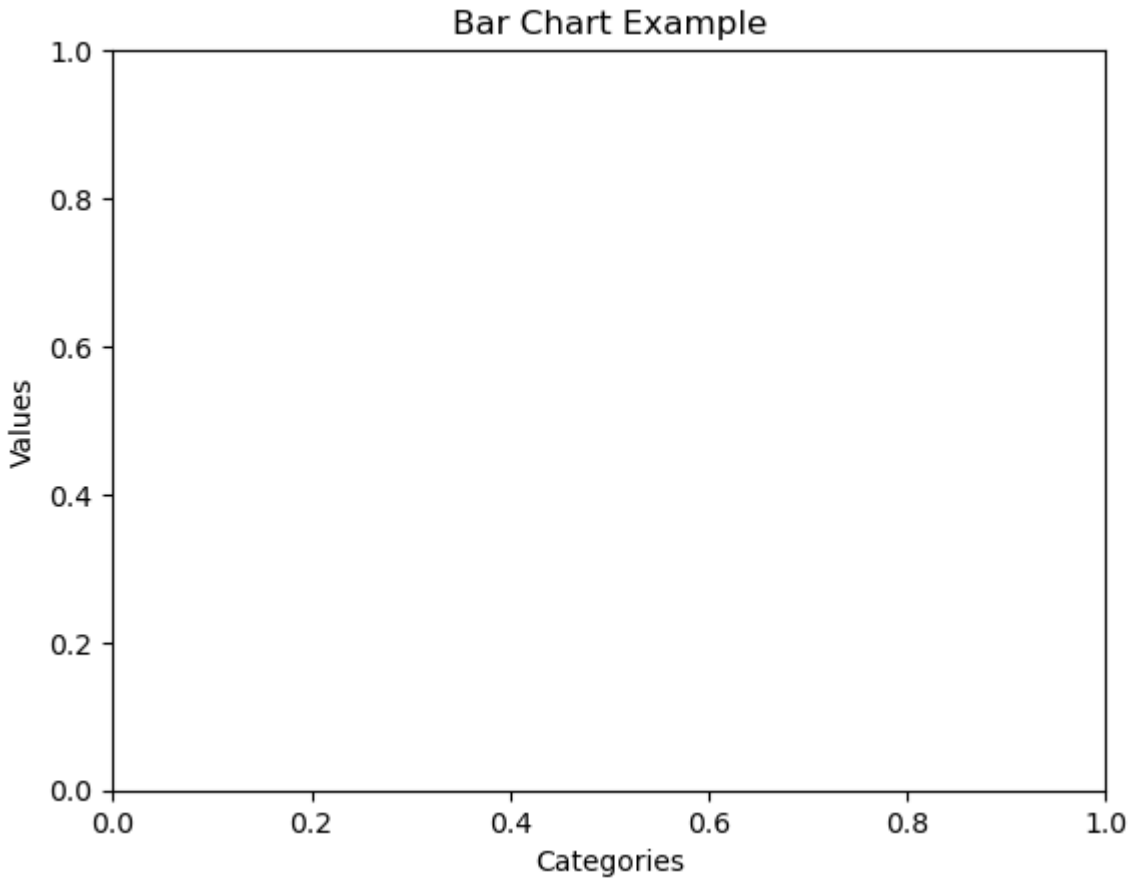
In [5]:
```
#//Example 1: Bar Chart
categories = ['Category A', 'Category B', 'Category C']
values = [25, 50, 30]

# Create bar chart
plt.bar(categories, values, color='blue')
```

Out[5]:
```
<BarContainer object of 3 artists>
```



In [6]:
```
# Add labels and title
plt.xlabel('Categories')
plt.ylabel('Values')
plt.title('Bar Chart Example')
```

Out[6]:
```
Text(0.5, 1.0, 'Bar Chart Example')
```

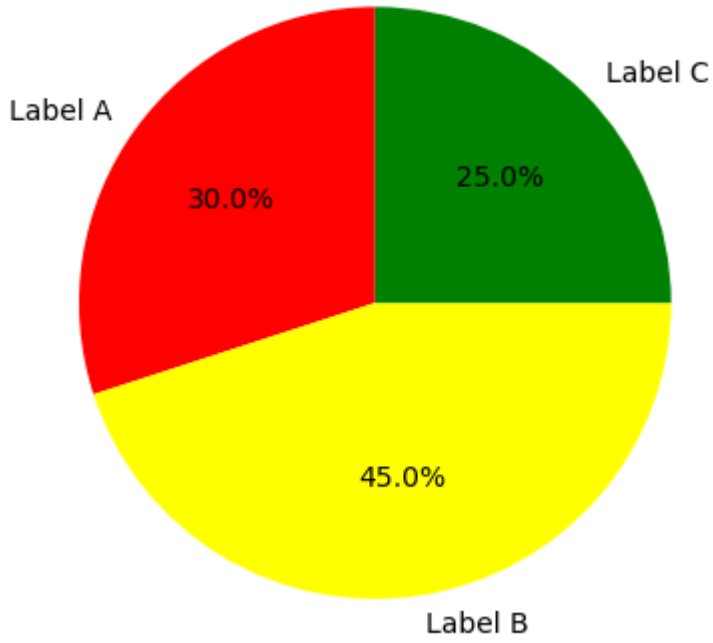


In [7]:
```
# Show the plot
plt.show()
```

In [9]:
```
### Example 2: Pie Chart
labels = ['Label A', 'Label B', 'Label C']
sizes = [30, 45, 25]
```

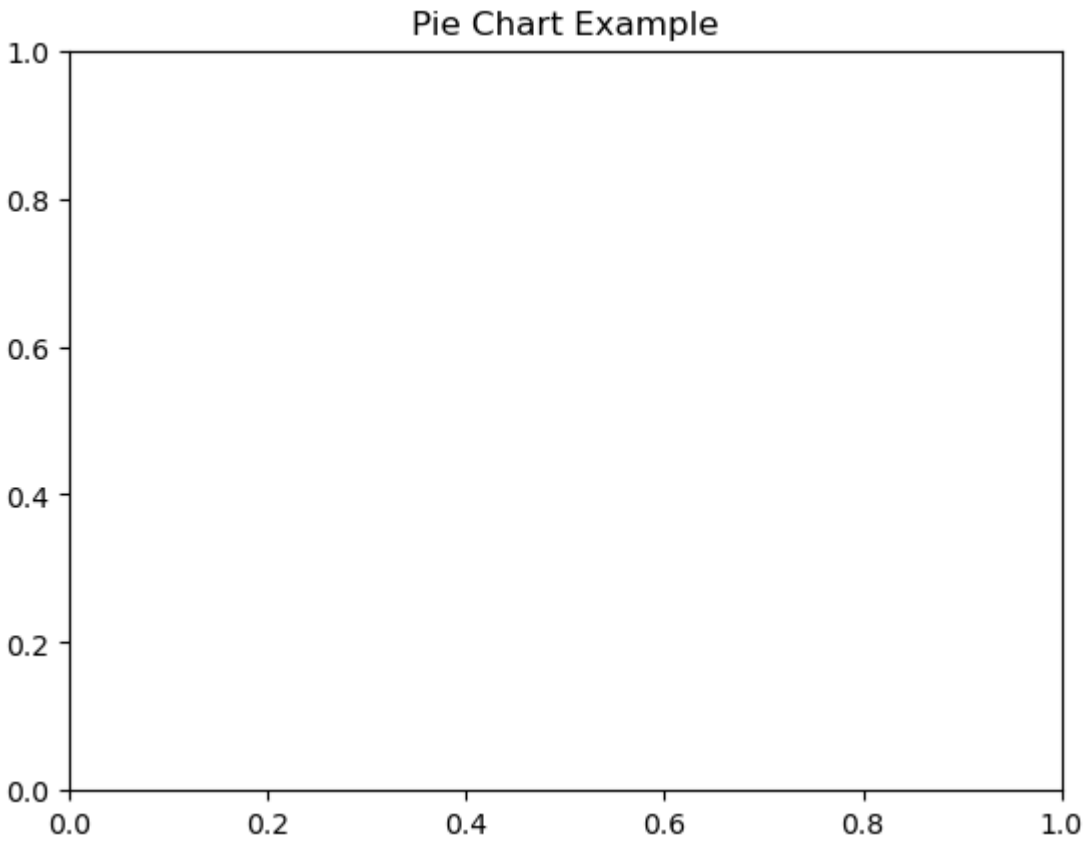
In [11]:
```
# Create pie chart
plt.pie(sizes, labels=labels, autopct='%1.1f%%', startangle=90, colors=['red', 'yellow
```

Out[11]:
```
([<matplotlib.patches.Wedge at 0x2179f4ae460>,
  <matplotlib.patches.Wedge at 0x2179f4aeb80>,
  <matplotlib.patches.Wedge at 0x2179f4bb2e0>],
 [Text(-0.8899187180267095, 0.6465637441936395, 'Label A'),
  Text(0.17207795223283862, -1.086457168210212, 'Label B'),
  Text(0.7778174593052025, 0.7778174593052022, 'Label C')],
 [Text(-0.48541020983275057, 0.3526711331965306, '30.0%'),
  Text(0.09386070121791196, -0.5926130008419338, '45.0%'),
  Text(0.4242640687119286, 0.4242640687119284, '25.0%')])
```



In [14]:
```
#Add title
plt.title('Pie Chart Example')
```

Out[14]:
```
Text(0.5, 1.0, 'Pie Chart Example')
```

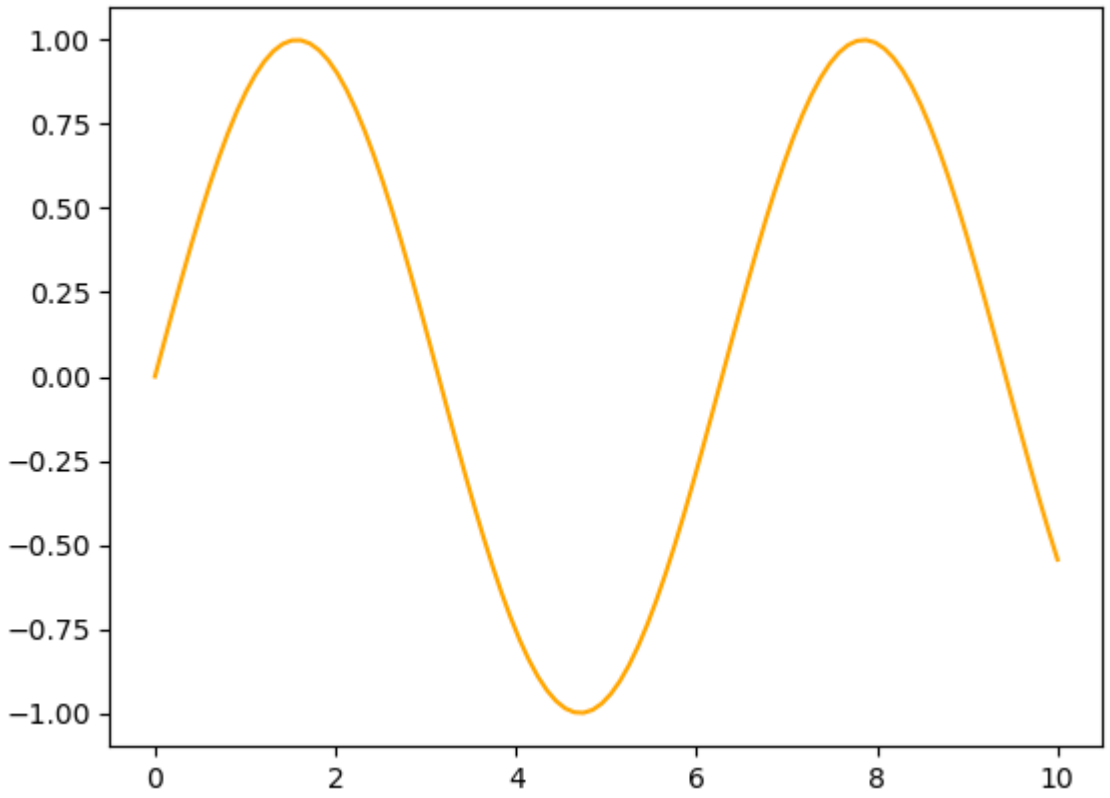


In [15]:
```
# Show the plot
plt.show()
```

In [16]:
```
### Example 3: Line Graph
x = np.linspace(0, 10, 100)
y = np.sin(x)
```

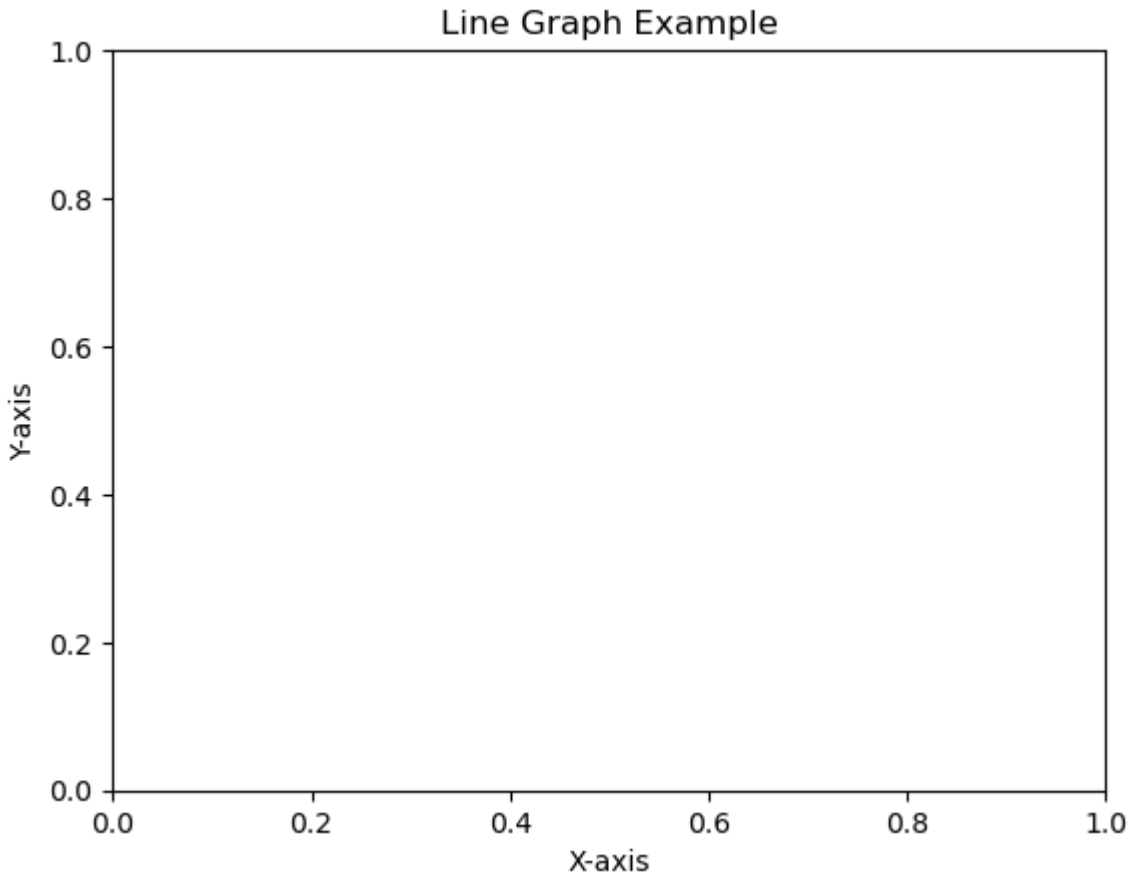
In [17]:
```
# Create line graph
plt.plot(x, y, label='sin(x)', color='orange')
```

Out[17]:
```
[<matplotlib.lines.Line2D at 0x2179f7f8d90>]
```



In [18]:
```
# Add labels and title
plt.xlabel('X-axis')
plt.ylabel('Y-axis')
plt.title('Line Graph Example')
```

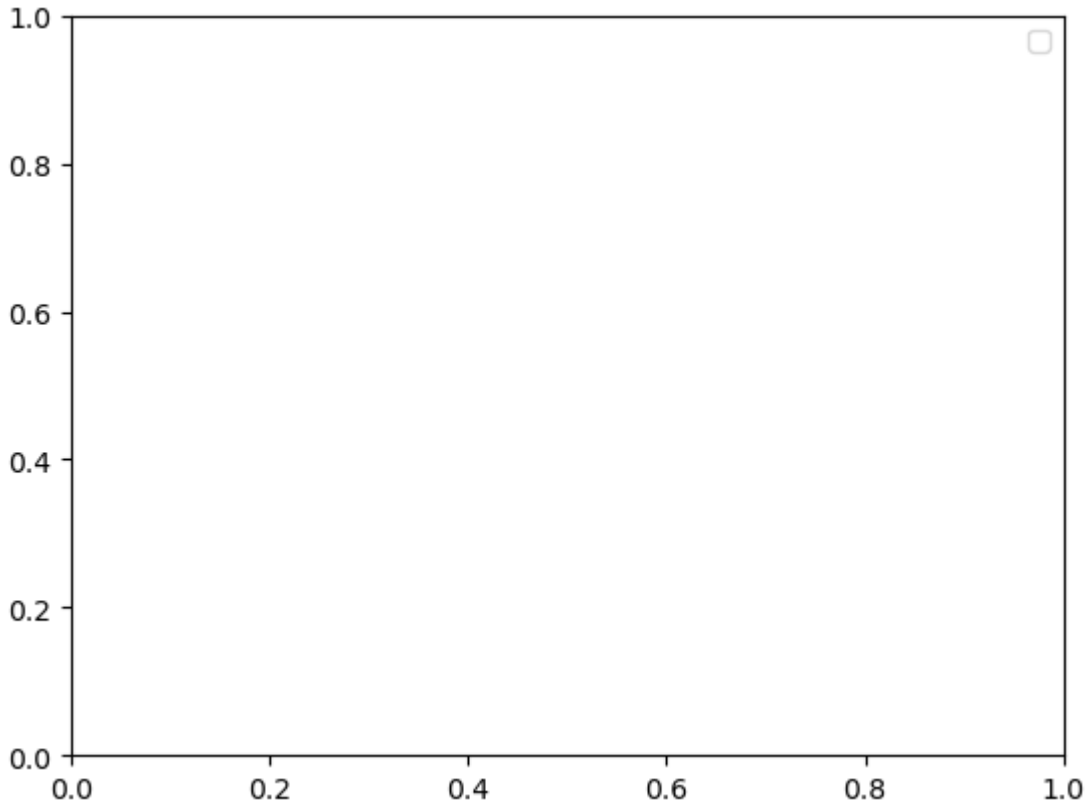
Out[18]:
```
Text(0.5, 1.0, 'Line Graph Example')
```



In [19]:
```
# Add a Legend
plt.legend()
```

```
No artists with labels found to put in legend.  Note that artists whose label start w
ith an underscore are ignored when legend() is called with no argument.
```

Out[19]:
```
<matplotlib.legend.Legend at 0x2179f707370>
```



In [20]:
```
# Show the plot
plt.show()
```

In [ ]:

In [1]:
```
import random
```

In [2]:
```
# Quiz questions and answers
quiz_data = {
    'General Knowledge': [
        {
            'question': 'What is the capital of France?',
            'options': ['Berlin', 'Paris', 'London', 'Rome'],
            'answer': 'Paris'
        },
        {
            'question': 'Which planet is known as the Red Planet?',
            'options': ['Mars', 'Venus', 'Jupiter', 'Saturn'],
            'answer': 'Mars'
        }
    ],
    'Programming': [
        {
            'question': 'What does HTML stand for?',
            'options': ['Hypertext Markup Language', 'Hyper Transfer Markup Language',
            'answer': 'Hypertext Markup Language'
        },
        {
            'question': 'In Python, how do you declare a variable?',
            'options': ['var x', 'int x', 'x = 5', 'declare x'],
            'answer': 'x = 5'
        }
    ]
}
```

In [3]:
```
# Function to display a question and get user input
def ask_question(category, question_data):
    print(f'\nCategory: {category}')
    print(question_data['question'])

    for i, option in enumerate(question_data['options'], start=1):
        print(f'{i}. {option}')

    user_answer = input('Your answer (enter the number): ')
    return question_data['options'][int(user_answer) - 1]
```

In [ ]:
```
# Function to conduct the quiz
def conduct_quiz():
    score = 0

    for category, questions in quiz_data.items():
        print('\n---', category, '---')

        random.shuffle(questions)  # Shuffle the questions in each category

        for question in questions:
            user_choice = ask_question(category, question)

            if user_choice == question['answer']:
                print('Correct!\n')
                score += 1
            else:
```

```
                print(f'Incorrect. The correct answer is: {question["answer"]}\n')

    print(f'\nQuiz completed! Your final score is: {score} out of {len(quiz_data["Gene

# Run the quiz
conduct_quiz()
```

```
--- General Knowledge ---

Category: General Knowledge
Which planet is known as the Red Planet?
1. Mars
2. Venus
3. Jupiter
4. Saturn
Your answer (enter the number): 2
Incorrect. The correct answer is: Mars


Category: General Knowledge
What is the capital of France?
1. Berlin
2. Paris
3. London
4. Rome
```

In [ ]:

In [1]:
```
import random
import string
```

In [3]:
```
def generate_password(length, include_letters=True, include_numbers=True, include_symb
    characters = ''

    if include_letters:
        characters += string.ascii_letters
    if include_numbers:
        characters += string.digits
    if include_symbols:
        characters += string.punctuation

    if not characters:
        print("Error: Please include at least one of letters, numbers, or symbols.")
        return None

    password = ''.join(random.choice(characters) for _ in range(length))
    return password

# Example usage:
length = int(input("Enter the desired password length: "))
include_letters = input("Include letters? (y/n): ").lower() == 'y'
include_numbers = input("Include numbers? (y/n): ").lower() == 'y'
include_symbols = input("Include symbols? (y/n): ").lower() == 'y'

password = generate_password(length, include_letters, include_numbers, include_symbols
if password:
    print("Generated Password:", password)
```

```
Enter the desired password length: 5
Include letters? (y/n): Y
Include numbers? (y/n): Y
Include symbols? (y/n): Y
Generated Password: (b^2l
```

In [ ]: