**TITLE:** CodTech IT Solutions Internship - Task Documentation using python

**INTERN INFORMATION:**

**Name:** SHAYLEE AMIRTHA R

**ID:** COD7094

**INTRODUCTION**

This program is designed to provide users with a versatile and user-friendly tool for performing basic arithmetic operations as well as advanced mathematical calculations such as exponentiation. Whether you need to add, subtract, multiply, divide, or find the power of a number, our calculator has got you covered.

With a simple and intuitive interface, users can easily input their desired operations and operands, and the calculator will promptly display the result. Additionally, the program offers the convenience of quitting at any time, making it a hassle-free tool for quick calculations.

In this comprehensive documentation, we will walk you through the functionality of our Python Simple Calculator, explaining how it works and providing detailed insights into its code structure. We will also include images illustrating the program's execution and user interactions, allowing you to visualize the calculator in action.

Task 02

Our chatbot is an intelligent conversational agent designed to engage with users in natural language and provide relevant information or assistance based on their queries. Powered by natural language processing techniques, our chatbot aims to understand user input accurately and respond contextually, contributing to a seamless and enjoyable user experience.

In today's digital era, chatbots have become increasingly popular for various applications, from customer service and support to personal assistants and entertainment. Our Python Chatbot offers a versatile platform for interacting with users, offering assistance, answering questions, or simply engaging in casual conversation.

This comprehensive documentation will guide you through the functionality of our Python Chatbot, explaining how it works and providing detailed insights into its code structure. We will also include images illustrating the program's execution and user interactions, allowing you to experience the chatbot in action firsthand.

**Implementation**

**Code 1:**

def add(x, y):

"""Function to add two numbers"""

return x + y

def subtract(x, y):

"""Function to subtract two numbers"""

return x - y

def multiply(x, y):

"""Function to multiply two numbers"""

return x \* y

def divide(x, y):

"""Function to divide two numbers"""

if y == 0:

return "Cannot divide by zero!"

return x / y

def exponentiate(x, y):

"""Function to calculate x raised to the power of y"""

return x \*\* y

def calculator():

print("Welcome to Simple Calculator!")

print("Select operation:")

print("1. Addition")

print("2. Subtraction")

print("3. Multiplication")

print("4. Division")

print("5. Exponentiation")

print("6. Quit")

while True:

choice = input("Enter choice (1/2/3/4/5/6): ")

if choice in ('1', '2', '3', '4', '5'):

num1 = float(input("Enter first number: "))

num2 = float(input("Enter second number: "))

if choice == '1':

print("Result:", add(num1, num2))

elif choice == '2':

print("Result:", subtract(num1, num2))

elif choice == '3':

print("Result:", multiply(num1, num2))

elif choice == '4':

print("Result:", divide(num1, num2))

elif choice == '5':

print("Result:", exponentiate(num1, num2))

elif choice == '6':

print("Thank you for using Simple Calculator!")

break

else:

print("Invalid input")

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

calculator()

**code 2:**

import random

# Define responses for the chatbot

responses = {

"hi": ["Hello!", "Hi there!", "Hey!"],

"how are you?": ["I am doing well, thank you!", "I'm good, thanks for asking. How about you?"],

"good": ["That's great to hear!", "Awesome!", "Good to know."],

"not good": ["I'm sorry to hear that. How can I help you?", "Is there anything I can do to make you feel better?"],

"thank you": ["You're welcome!", "No problem.", "Anytime!"],

"bye": ["Goodbye!", "Take care!", "See you later!"],

"default": ["I'm not sure I understand. Could you please rephrase that?"]

}

def get\_response(user\_input):

"""Get response based on user input"""

for pattern, responses\_list in responses.items():

if pattern in user\_input.lower():

return random.choice(responses\_list)

return random.choice(responses["default"])

def chat():

"""Start the conversation with the chatbot"""

print("Welcome! How can I assist you today?")

while True:

user\_input = input("> ")

if user\_input.lower() == 'quit':

print("Goodbye!")

break

else:

response = get\_response (user\_input)

print(response)

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

chat()

**CODE EXPLAINATION:**

**Code 1:**

The program defines functions for basic arithmetic operations: add, subtract, multiply, divide, and exponentiate.

The calculator() function presents a menu of operations to the user and prompts for input.

Inside a while loop, the program continuously asks the user for their choice of operation until they choose to quit.

Depending on the user's choice, the program calls the corresponding function to perform the operation on two numbers entered by the user.

The program validates user input and handles division by zero error.

**Code 2 :**

The program begins by importing the necessary libraries. In this case, we import the random module for generating random responses.

We define a dictionary named responses to map user queries (keys) to corresponding lists of responses. Each key represents a pattern that the chatbot recognizes in the user input, and each value is a list of possible responses.

The get\_response() function takes the user input and searches for matching patterns in the responses dictionary. If a match is found, it returns a randomly chosen response from the corresponding list. If no match is found, it returns a default response.

The chat() function initiates the conversation with the user. It continuously prompts the user for input until the user types "quit".

User input is processed by the get\_response() function, and the appropriate response is printed.

**Python Functionality:**

The functionality of the Python Chatbot revolves around its ability to understand user input and generate contextually relevant responses. Here's a breakdown of the key functionality:

1. Natural Language Understanding: The chatbot employs natural language processing (NLP) techniques to understand user queries. It analyzes the input text to identify patterns and keywords that correspond to specific intents or topics.

2. Response Generation: Based on the user input, the chatbot selects an appropriate response from a predefined set of responses. These responses are designed to be contextually relevant and can vary depending on the detected intent or topic of the user query.

3. Pattern Matching: The chatbot uses pattern matching to recognize specific phrases or keywords in the user input. This allows it to identify common greetings, inquiries about the chatbot's well-being, expressions of gratitude, farewells, and more.

4. Randomization: To enhance the conversational experience, the chatbot incorporates randomness in its response selection. When multiple responses are available for a given pattern, the chatbot randomly selects one to add variety and make the interaction more engaging.

5. User Interaction: The chatbot engages in a continuous conversation with the user, prompting for input and responding accordingly. It maintains the conversation flow until the user decides to end the interaction by typing "quit".

**USAGE:**

1. Starting the Chatbot: To start the Python Chatbot, simply run the Python script containing the chatbot code. You can do this by executing the script in your preferred Python environment, such as IDLE, command line, or a code editor.

python chatbot.py

2. Initiating Conversation: Once the chatbot is running, it will display a welcome message prompting the user to enter their query or message.

Welcome! How can I assist you today?

3. Interacting with the Chatbot: Enter your message or query after the prompt, and the chatbot will respond accordingly. You can engage in a conversation by asking questions, making statements, or expressing sentiments.

> Hi

Hello!

> How are you?

I am doing well, thank you!

> What can you do?

I can provide information, answer questions, or just have a chat with you. Feel free to ask!

> Thank you

You're welcome!

> Quit

Goodbye!

4. Ending the Conversation: To exit the chatbot and end the conversation, simply type "quit" or any other predefined exit command. This will prompt the chatbot to display a farewell message and terminate the program.

> quit

Goodbye!

5. Handling User Input: The chatbot is designed to understand natural language input, so you can interact with it using conversational language. It utilizes pattern matching and natural language processing techniques to interpret user queries and generate contextually relevant responses.

6. Enjoying the Interaction: Engage in a conversation with the chatbot, ask questions, seek assistance, or simply enjoy the interaction. The chatbot is here to provide relevant information and assistance, contributing to a seamless user experience.

**CONCLUSION**

In conclusion, both the Python Simple Calculator and Chatbot projects offer versatile solutions for different purposes. The Simple Calculator provides users with a straightforward tool for performing basic arithmetic operations as well as advanced calculations like exponentiation, enhancing efficiency in mathematical tasks. On the other hand, the Chatbot utilizes natural language processing techniques to engage in conversations with users, providing relevant information and assistance based on their queries. Both projects demonstrate the power of Python in developing interactive and functional applications, catering to diverse user needs and contributing to a seamless user experience in different contexts**.**

**OUTPUT:**



