## Experiment No. 12

**Aim:** Make a simple program that can talk to you and answer questions. You can start with basic rules and then try to make it smarter by teaching it new things.

**Course Outcome:** Develop simple Chatbot.

**Theory:**

**What is a Chatbot?**

A chatbot is a software application designed to simulate human conversation through text or voice. At its core, it's a program that maps a user's input (what you say) to an output (what the bot says back). The "magic" is in how it makes that map.

**Approach 1: The Rule-Based Chatbot (The Simple Start)**

This is the easiest way to start and works like a simple flowchart. You, the programmer, create a set of explicit rules.

Theory: Keyword Matching and If-Else Logic

The chatbot's entire "brain" is a collection of if-then or if-else statements. It doesn't understand the meaning of a sentence; it only recognizes keywords or patterns.

1. Scan for Keywords: The program takes the user's input (e.g., "What is the time?") and scans it for predefined keywords ("time").
2. Match a Rule: It checks its list of rules. Does the input contain "hello" or "hi"? If yes, then respond with "Hello there!". Does the input contain "time"? If yes, then respond with the current time.
3. Provide a Default Response: If the input contains no recognizable keywords, the chatbot gives a generic fallback answer like, "Sorry, I don't understand that."

Think of it like a vending machine. You press a specific button (the keyword), and it gives you a specific, pre-loaded snack (the response). It can't handle a request like "I'm feeling hungry for something salty but not chips." Example Logic:

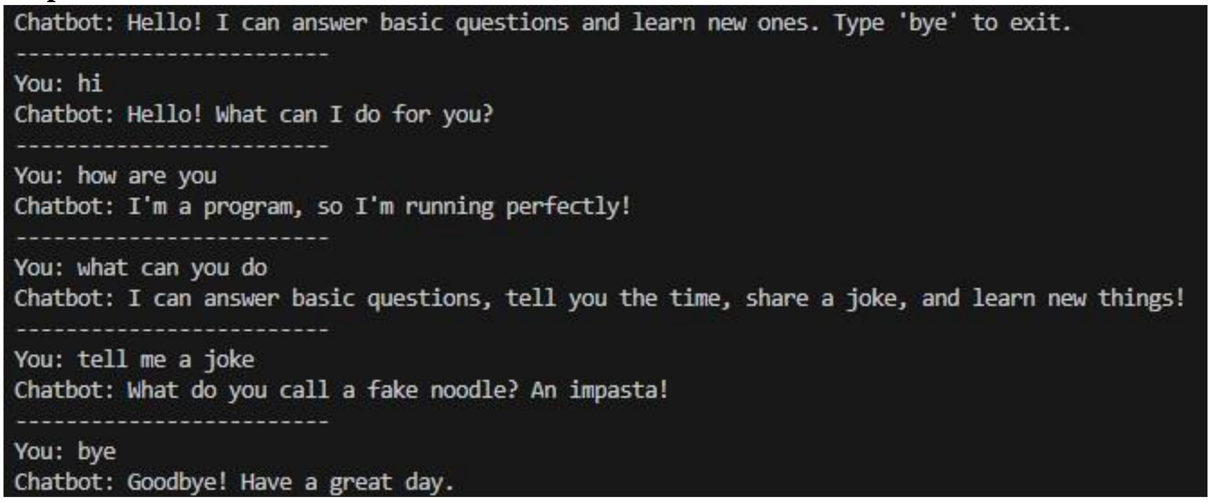
- IF user input contains "hello" THEN reply "Hi! How can I help you?"
- ELSE IF user input contains "help" THEN reply "I can answer questions about our hours and location."
- ELSE IF user input contains "hours" THEN reply "We are open from 9 AM to 5 PM."
- ELSE reply "I'm not sure how to answer that."

**Program:**

```
import json import random

from datetime import datetime


class Chatbot:     def __init__(self,

knowledge_base_file: str):
        self.file_path = knowledge_base_file

self.knowledge_base = self._load_knowledge_base()

        self.jokes = [
            "Why don't scientists trust atoms? Because they make up everything!",
            "What do you call a fake noodle? An impasta!",
            "Why did the scarecrow win an award? Because he was outstanding in his field!"
        ]

    def _load_knowledge_base(self) -> dict:
        try:            with open(self.file_path,

'r') as file:

                return json.load(file)
        except (FileNotFoundError, json.JSONDecodeError):
            return {
                "hello": "Hi there! How can I help you?",
                "hi": "Hello! What can I do for you?",
                "hey": "Hey! How's it going?",
                "how are you": "I'm a program, so I'm running perfectly!",
                "what are you": "I am a simple, rule-based chatbot.",
                "what can you do": "I can answer basic questions, tell you the time, share a joke,
and learn new things!",
                "what is your name": "You can call me XD.",
                "who made you": "I was created by a developer using Python.",
                "what is python": "Python is a high-level, general-purpose programming
language.",
                "thanks": "You're welcome!",
                "thank you": "You're welcome! Happy to help.",
                "bye": "Goodbye! Have a great day.",
                "goodbye": "Talk to you later!"
            }

    def _save_knowledge_base(self):

with open(self.file_path, 'w') as file:

            json.dump(self.knowledge_base, file, indent=4)

    def get_response(self, user_input: str) -> str:

user_input_lower = user_input.lower().strip()


        if "time" in user_input_lower:
         now = datetime.now()            return f"The current time is

{now.strftime('%I:%M %p')}. (IST)"


      if "date" in user_input_lower:            today = datetime.now()

return f"Today's date is {today.strftime('%A, %B %d, %Y')}."


      if "joke" in user_input_lower:
         return random.choice(self.jokes)

      if user_input_lower in self.knowledge_base:
         return self.knowledge_base[user_input_lower]

return None


   def learn(self, question: str, answer: str):        self.knowledge_base[question.lower().strip()]

= answer.strip()        self._save_knowledge_base()        print("Chatbot: Thanks! I've stored

that for next time.")


   def start_chat(self):
      print("Chatbot: Hello! I can answer basic questions and learn new ones. Type 'bye' to
exit.")        print("-" * 25)

      while True:
         user_message = input("You: ")

         if user_message.lower().strip() in ["bye", "goodbye"]:
            print(f"Chatbot: {self.knowledge_base.get(user_message.lower().strip(),
'Goodbye!')}")
            break

         response = self.get_response(user_message)

         if response:
            print(f"Chatbot: {response}")
         else:
            print("Chatbot: Hmm, I don't know the answer to that. Can you teach me?")

new_answer = input(f"What is the correct response for '{user_message}'? \nYou: ")
```

```
                if new_answer.lower().strip() not in ['skip', 'no']:
                    self.learn(user_message, new_answer)
                else:
                    print("Chatbot: Okay, I'll skip that one.")

            print("-" * 25)
if __name__ == "__main__":
    bot = Chatbot(knowledge_base_file="bot_memory.json")
    bot.start_chat()
```

**Output:**



```
Chatbot: Hello! I can answer basic questions and learn new ones. Type 'bye' to exit.
-------------------------
You: hi
Chatbot: Hello! What can I do for you?
-------------------------
You: how are you
Chatbot: I'm a program, so I'm running perfectly!
-------------------------
You: what can you do
Chatbot: I can answer basic questions, tell you the time, share a joke, and learn new things!
-------------------------
You: tell me a joke
Chatbot: What do you call a fake noodle? An impasta!
-------------------------
You: bye
Chatbot: Goodbye! Have a great day.
```

**Conclusion:** Successfully created a simple program that can talk to you and answer questions.