**Assignment 9**

**Problem Statement:**  
Create a **Chatbot application** for any real-world scenario.

**Theory**

**1. Chatbots in AI**

* A **chatbot** is a conversational agent that interacts with users using natural language.
* They can be **rule-based** (keyword matching, predefined responses) or **AI-driven** (machine learning, NLP).
* Applications: customer support, healthcare assistants, virtual shopping guides, educational tutors.

**2. Rule-Based Chatbot**

* Works using **pattern matching** and **conditional logic**.
* For each input keyword/phrase, the bot responds with a predefined message.
* Simple to implement but limited in handling complex conversations.

**3. Real-World Scenario – Hospital Assistant Chatbot**

* A hospital chatbot can:
  + Greet users
  + Help book appointments
  + Provide doctor availability
  + Share hospital timings
  + Provide emergency information

**Algorithm**

1. Take user input.
2. Convert input to lowercase for easy matching.
3. Check if input contains certain **keywords** (e.g., “hello”, “appointment”, “doctor”, “emergency”).
4. Respond with the corresponding predefined reply.
5. Continue until user types “bye” or “exit”.

**Code (C++ Implementation)**

#include <iostream>

#include <string>

#include <algorithm>

using namespace std;

// Convert string to lowercase for easy comparison

string toLower(string s) {

transform(s.begin(), s.end(), s.begin(), ::tolower);

return s;

}

int main() {

cout << "==============================\n";

cout << " Hospital Assistant Chatbot\n";

cout << "==============================\n";

cout << "Type 'bye' to exit.\n\n";

string userInput;

while (true) {

cout << "You: ";

getline(cin, userInput);

string msg = toLower(userInput);

if (msg == "bye" || msg == "exit") {

cout << "Chatbot: Goodbye! Take care.\n";

break;

}

else if (msg.find("hello") != string::npos || msg.find("hi") != string::npos) {

cout << "Chatbot: Hello! How can I help you today?\n";

}

else if (msg.find("appointment") != string::npos) {

cout << "Chatbot: Sure! Please provide your preferred date and department.\n";

}

else if (msg.find("doctor") != string::npos) {

cout << "Chatbot: We have doctors available in General Medicine, Pediatrics, and Dermatology.\n";

}

else if (msg.find("time") != string::npos || msg.find("hours") != string::npos) {

cout << "Chatbot: Our hospital is open from 9 AM to 8 PM, Monday to Saturday.\n";

}

else if (msg.find("emergency") != string::npos) {

cout << "Chatbot: In case of emergency, please dial 108 immediately!\n";

}

else if (msg.find("thank") != string::npos) {

cout << "Chatbot: You're welcome! Anything else I can help you with?\n";

}

else {

cout << "Chatbot: I'm sorry, I didn't understand that. Can you rephrase?\n";

}

}

return 0;

}

**Sample Output**

==============================

Hospital Assistant Chatbot

==============================

Type 'bye' to exit.

You: hello

Chatbot: Hello! How can I help you today?

You: I want to book an appointment

Chatbot: Sure! Please provide your preferred date and department.

You: what time do you open?

Chatbot: Our hospital is open from 9 AM to 8 PM, Monday to Saturday.

You: thanks

Chatbot: You're welcome! Anything else I can help you with?

You: bye

Chatbot: Goodbye! Take care.

**Conclusion**

* A **rule-based chatbot** was successfully implemented for a **Hospital Assistant** scenario.
* The chatbot can:
  + Greet users
  + Book appointments
  + Provide doctor details
  + Share hospital timings
  + Handle emergencies
* While simple, this demonstrates the foundation of chatbot applications.
* In real-world applications, chatbots can be enhanced using **Natural Language Processing (NLP)** and **Machine Learning** for smarter, context-aware responses.