class ExpertSystem:

def \_\_init\_\_(self):

self.rules = {

'critical\_condition': {

'symptoms': ['difficulty\_breathing', 'unconsciousness', 'severe\_bleeding'],

'response': 'I'm sorry to hear that. This sounds like a critical condition. Please call emergency services immediately.'

},

'fever': {

'symptoms': ['fever'],

'response': 'It seems like the patient has a fever. It is advisable to consult with the general medicine department.'

},

'chest\_pain': {

'symptoms': ['chest\_pain'],

'response': 'Chest pain can be a serious symptom. Please consult with the cardiology department.'

},

# Add more rules for different conditions and departments

}

def infer\_department(self, symptoms):

for condition, data in self.rules.items():

if any(symptom in data['symptoms'] for symptom in symptoms):

return condition

return None

class HospitalChatbot:

def \_\_init\_\_(self, expert\_system):

self.expert\_system = expert\_system

def chat(self):

print("Hospital Management System Chatbot: Hello! How can I assist you today?")

while True:

user\_input = input("You: ").lower()

if user\_input == 'exit':

print("Hospital Management System Chatbot: Goodbye! Take care.")

break

symptoms = user\_input.split(',')

department = self.expert\_system.infer\_department(symptoms)

if department:

print(f"Hospital Management System Chatbot: {self.expert\_system.rules[department]['response']}")

else:

print("Hospital Management System Chatbot: I'm sorry, I didn't understand. Can you please provide more information?")

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

expert\_system = ExpertSystem()

hospital\_chatbot = HospitalChatbot(expert\_system)

hospital\_chatbot.chat()