Lab 3

1. Simple Reflex Agents:

These agents select actions based on the current percept (what they perceive from the environment). They do not store any history of previous states.

Example:

- **Thermostat:** A thermostat monitors the room temperature (percept) and turns the heater on or off (action) based on the current temperature compared to a desired set point.

2. Model-Based Reflex Agents:

These agents maintain some internal state that helps them keep track of aspects of the environment that may change or be hidden, allowing them to make more informed decisions.

Example:

- **Self-Driving Car:** The car uses sensors (cameras, LIDAR, etc.) to perceive the environment and maintains an internal state that includes information like the car's speed, location of other vehicles, and traffic signals. The car makes decisions like turning, stopping, or accelerating based on both the current and previous information.

3. Goal-Based Agents:

These agents act to achieve specific goals. They use the current state of the environment and their goals to decide the best action.

Example:

- **Robot Vacuum Cleaner:** The robot vacuum uses its sensors to map the room and set a goal (e.g., clean the entire floor). It plans a path and moves around obstacles to achieve the goal of cleaning the entire room.

4. Utility-Based Agents:

These agents act not just to achieve goals but also to maximize a "utility" function (a measure of satisfaction). They evaluate different possible actions and select the one that maximizes their utility.

Example:

- **Investment AI:** A financial AI agent that trades stocks based on maximizing expected returns. It evaluates different investment opportunities (actions) and selects the ones that provide the highest expected utility (profit).

5. Learning Agents:

These agents have the ability to learn from their environment, experiences, and feedback. They improve their performance over time by adjusting their actions based on past results.

Example:

- **Chatbot:** A chatbot designed to assist customers learns from each interaction. Initially, it might only handle simple queries, but over time, it improves its responses based on user feedback, becoming more effective at answering questions.

6. Multi-Agent Systems:

In these systems, multiple agents interact or collaborate to solve problems that might be too complex for a single agent. Each agent may have its own goals, but they work together to achieve a common objective.

Example:

- **Autonomous Drone Fleet:** In search and rescue operations, multiple drones (agents) might be deployed to survey a large area. Each drone gathers information and communicates with others to cover different regions effectively, optimizing the search process.

Scenario: Smart Home Temperature Control

Description:

Imagine a smart home with multiple rooms. Each room has a thermostat that controls the heating system. The goal is to maintain a comfortable temperature in each room (e.g., 22°C). The smart home system is equipped with a Simple Reflex Agent that can:

- Sense the current temperature of each room.
- Turn the heater on if the temperature is below 22°C.
- Turn the heater off if the temperature is 22°C or above.

We will also explore a Model-Based Reflex Agent that remembers whether the heater was on or off in the previous cycle to avoid redundant actions.

Simple Reflex Agent Example:

This agent will check the current temperature and decide whether to turn the heater on or off.

```
class SimpleReflexAgent:
    def __init__(self, desired_temperature):
        self.desired_temperature = desired_temperature

    def perceive(self, current_temperature):
        return current_temperature

    def act(self, current_temperature):
        if current_temperature < self.desired_temperature:
            action = "Turn on heater"
        else:
            action = "Turn off heater"
        return action

# simulating different rooms with different current temperatures
rooms = {
    "Living Room": 18,
    "Bedroom": 22,
    "Kitchen": 20,
    "Bathroom": 24
}

# desired temperature for all rooms
desired_temperature = 22
agent = SimpleReflexAgent(desired_temperature)

# run the agent for each room
for room, temperature in rooms.items():
    action = agent.act(temperature)
    print(f"{room}: Current temperature = {temperature}°C. {action}.")
```

Lab 3 Task:

- **Model-Based Reflex Agent**
- This agent not only checks the current temperature but also remembers the previous action to avoid turning the heater on or off unnecessarily.