Exp 1

Implementation of Reinforcement Learning System: Agent, Environment, State, Actions, and Reward.

Code 1

```
import random

class Environment:
    def __init__(self):  # Fixed the init method
        self.state = 0

    def reset(self):
        self.state = 0
        return self.state

    def step(self, action):
        self.state += action
        if self.state == 5:
            reward = 10
            done = True
        else:
            reward = 0
            done = False
        return self.state, reward, done

class Agent:
    def __init__(self):  # Fixed the init method
        self.actions = [-1, +1]

    def select_action(self):
        return random.choice(self.actions)  # Fixed self.actions

# Initialize the environment and agent
env = Environment()
agent = Agent()

state = env.reset()
done = False
step_count = 0

print("Starting Episodes")

while not done:
    action = agent.select_action()
    next_state, reward, done = env.step(action)
```

```
    print(f"step {step_count}; State={state}, Action={action}, Next
State={next_state}, Reward={reward}")
    state = next_state
    step_count += 1

print("Episode Finished")
```

Output

```
Starting Episodes
step 0; State=0, Action=-1, Next State=-1, Reward=0
step 1; State=-1, Action=1, Next State=0, Reward=0
step 2; State=0, Action=1, Next State=1, Reward=0
step 3; State=1, Action=1, Next State=2, Reward=0
step 4; State=2, Action=-1, Next State=1, Reward=0
step 5; State=1, Action=1, Next State=2, Reward=0
step 6; State=2, Action=-1, Next State=1, Reward=0
step 7; State=1, Action=-1, Next State=0, Reward=0
step 8; State=0, Action=-1, Next State=-1, Reward=0
step 9; State=-1, Action=-1, Next State=-2, Reward=0
step 10; State=-2, Action=1, Next State=-1, Reward=0
step 11; State=-1, Action=-1, Next State=-2, Reward=0
step 12; State=-2, Action=-1, Next State=-3, Reward=0
step 13; State=-3, Action=1, Next State=-2, Reward=0
step 14; State=-2, Action=1, Next State=-1, Reward=0
step 15; State=-1, Action=1, Next State=0, Reward=0
step 16; State=0, Action=-1, Next State=-1, Reward=0
step 17; State=-1, Action=-1, Next State=-2, Reward=0
step 18; State=-2, Action=-1, Next State=-3, Reward=0
step 19; State=-3, Action=1, Next State=-2, Reward=0
step 20; State=-2, Action=1, Next State=-1, Reward=0
step 21; State=-1, Action=-1, Next State=-2, Reward=0
step 22; State=-2, Action=1, Next State=-1, Reward=0
step 23; State=-1, Action=1, Next State=0, Reward=0
step 24; State=0, Action=-1, Next State=-1, Reward=0
step 25; State=-1, Action=1, Next State=0, Reward=0
step 26; State=0, Action=-1, Next State=-1, Reward=0
step 27; State=-1, Action=1, Next State=0, Reward=0
step 28; State=0, Action=1, Next State=1, Reward=0
step 29; State=1, Action=1, Next State=2, Reward=0
step 30; State=2, Action=1, Next State=3, Reward=0
step 31; State=3, Action=-1, Next State=2, Reward=0
step 32; State=2, Action=-1, Next State=1, Reward=0
step 33; State=1, Action=-1, Next State=0, Reward=0
step 34; State=0, Action=1, Next State=1, Reward=0
step 35; State=1, Action=1, Next State=2, Reward=0
step 36; State=2, Action=-1, Next State=1, Reward=0
step 37; State=1, Action=1, Next State=2, Reward=0
step 38; State=2, Action=-1, Next State=1, Reward=0
step 39; State=1, Action=1, Next State=2, Reward=0
step 40; State=2, Action=1, Next State=3, Reward=0
step 41; State=3, Action=-1, Next State=2, Reward=0
step 42; State=2, Action=-1, Next State=1, Reward=0
```

```
step 43; State=1, Action=-1, Next State=0, Reward=0
step 44; State=0, Action=1, Next State=1, Reward=0
step 45; State=1, Action=1, Next State=2, Reward=0
step 46; State=2, Action=-1, Next State=1, Reward=0
step 47; State=1, Action=1, Next State=2, Reward=0
step 48; State=2, Action=-1, Next State=1, Reward=0
step 49; State=1, Action=1, Next State=2, Reward=0
step 50; State=2, Action=-1, Next State=1, Reward=0
step 51; State=1, Action=-1, Next State=0, Reward=0
step 52; State=0, Action=1, Next State=1, Reward=0
step 53; State=1, Action=-1, Next State=0, Reward=0
step 54; State=0, Action=-1, Next State=-1, Reward=0
step 55; State=-1, Action=-1, Next State=-2, Reward=0
step 56; State=-2, Action=-1, Next State=-3, Reward=0
```

Code 2

```python
import random

# Define the Environment
class Environment:
    def __init__(self):
        self.state = 0  # Starting position

    def reset(self):
        self.state = 0
        return self.state

    def step(self, action):
        self.state += action
        # Ensure the state doesn't go below 0
        if self.state < 0:
            self.state = 0

        # Define reward logic
        if self.state == 5:
            reward = 10
            done = True
        else:
            reward = 0
            done = False

        return self.state, reward, done

# Define the Agent
class Agent:
    def __init__(self):
        self.actions = [-1, 1]  # Move left or right
```

```
    def select_action(self):
        return random.choice(self.actions)

# Main logic
if __name__ == "__main__":
    env = Environment()
    agent = Agent()
    state = env.reset()
    done = False
    step_count = 0
    print("Starting Episode...")
    print("Agent starts at position 0. Goal is to reach position 5.\n")
    while not done:
        action = agent.select_action()
        next_state, reward, done = env.step(action)
        print(f"Step {step_count}: State={state}, Action={action}, Next
State={next_state}, Reward={reward}")
        state = next_state
        step_count += 1
    print("\nEpisode Finished. Agent reached position 5.")
```

Output

```
REINFORCEMENT LEARNING LAB EXPERIMENT 1
Starting Episode...
Agent starts at position 0. Goal is to reach position 5.

Step 0: State=0, Action=1, Next State=1, Reward=0
Step 1: State=1, Action=1, Next State=2, Reward=0
Step 2: State=2, Action=1, Next State=3, Reward=0
Step 3: State=3, Action=-1, Next State=2, Reward=0
Step 4: State=2, Action=1, Next State=3, Reward=0
Step 5: State=3, Action=1, Next State=4, Reward=0
Step 6: State=4, Action=1, Next State=5, Reward=10

Episode Finished. Agent reached position 5.
```