

```
[1]: class BlocksWorld:
    def __init__(self):
        # Initial state of blocks
        self.state = {
            "A": "B", # A is on B
            "B": "table", # B is on the table
            "C": "table" # C is on the table
        }
        # Goal state
        self.goal = {
            "A": "B",
            "B": "C",
            "C": "table"
        }

    def is_goal_state(self):
        """Checks if the current state matches the goal state."""
        return self.state == self.goal

    def move(self, block, destination):
        """Moves a block to a new destination if required."""
        if block in self.state and self.state[block] != destination:
            print(f"Moving {block} from {self.state[block]} to {destination}")
            self.state[block] = destination
```

```
"""Moves a block to a new destination if required."""
if block in self.state and self.state[block] != destination:
    print(f"Moving {block} from {self.state[block]} to {destination}")
    self.state[block] = destination

def plan_moves(self):
    """Executes moves step-by-step until the goal state is reached."""
    print("\nInitial State:", self.state)

    while not self.is_goal_state():
        for block, target in self.goal.items():
            if self.state[block] != target:
                self.move(block, target)

    print("\nFinal Goal State Reached:", self.state)

# Run the Blocks World Solver
bw = BlocksWorld()
bw.plan_moves()
```

Initial State: {'A': 'B', 'B': 'table', 'C': 'table'}  
Moving B from table to C  
  
Final Goal State Reached: {'A': 'B', 'B': 'C', 'C': 'table'}