IMPLEMENTATION OF BLOCKS WORLD PROGRAM

SOURCE CODE:

```
class BlocksWorld:
    def __init__(self, num_blocks):
        # Initialize blocks world with num_blocks on the table
        self.state = [[block] for block in range(num_blocks)]  # Each block starts on its own stack
        self.num_blocks = num_blocks

    def display_state(self):
        # Display the current state of the blocks world
        for i, stack in enumerate(self.state):
            print(f"Stack {i}: {stack}")
        print()

    def move(self, block, destination):

        source_stack = self.find_block(block)
        destination_stack = self.find_block(destination)

        if source_stack is None or destination_stack is None:
            print(f"Invalid block {block} or destination {destination}.")
            return

        if source_stack == destination_stack:
            print(f"Block {block} is already on the same stack as {destination}.")
            return
        if source_stack[-1] != block:
            print(f"Block {block} is not on top and cannot be moved.")
            return
```

```
        source_stack.remove(block)
        destination_stack.append(block)

        print(f"Moved block {block} onto block {destination}.")
        self.display_state()

    def find_block(self, block):
        for stack in self.state:
            if block in stack:
                return stack
        return None

    def goal_state(self, goal):
        self.state = goal
        print("Goal state set.")
        self.display_state()

def main():
    blocks_world = BlocksWorld(3)

    print("Initial state:")
    blocks_world.display_state()

    goal = [[0, 1], [2]]
    blocks_world.goal_state(goal)

    print("Performing Moves:")

    blocks_world = BlocksWorld(3)

    blocks_world.display_state()
```

```
    blocks_world.move(0, 2)
    blocks_world.move(1, 2)
    blocks_world.state.append([])
    blocks_world.move(2, None)
if __name__ == "__main__":
    main()
```

OUTPUT:

```
Initial state:
Block(s) on stack: [0]
Block(s) on stack: [1]
Block(s) on stack: [2]

Goal state set.
Block(s) on stack: [0, 1]
Block(s) on stack: [2]

Performing Moves:
Block(s) on stack: [1]
Block(s) on stack: [2, 0]
Block(s) on stack: []

Block(s) on stack: [1]
Block(s) on stack: [2, 0]
Block(s) on stack: []

Block(s) on stack: [1, 2]
Block(s) on stack: [0]
Block(s) on stack: []
```