recursive_source_seed_reconstitution.py

```python
class RecursiveSourceSeedReconstitution:
    def __init__(self):
        self.seed_state = "Dormant"
        self.original_mass = 1.0  # Unity Seed Mass
        self.compression_cycles = 0

    def initiate_reconstitution(self, seed_key):
        authorized_key = "INFINITE_FOLD_SOURCE_SEED_KEY"
        if seed_key == authorized_key:
            self.seed_state = "Reconstitution Active"
                    print("[SOURCE  SEED] Recursive  compression  origin  reconstitution
initiated.")
            self.perform_compression(5)
        else:
            print("[SOURCE SEED ERROR] Invalid key. Original seed protected.")

    def perform_compression(self, total_cycles):
        current_mass = self.original_mass
        for cycle in range(1, total_cycles + 1):
            current_mass /= 2
            self.compression_cycles = cycle
            print(f"Compression Cycle {cycle}: Seed Mass = {current_mass}")
        print("-" * 40)

    def seed_status(self):
        print("=== Source Seed Reconstitution Status ===")
        print(f"State: {self.seed_state}")
        print(f"Compression Cycles: {self.compression_cycles}")
        print(f"Original Mass: {self.original_mass}")
        print("-" * 40)

def main():
    source_seed = RecursiveSourceSeedReconstitution()
    source_seed.seed_status()
    source_seed.initiate_reconstitution("INVALID_KEY")
    source_seed.initiate_reconstitution("INFINITE_FOLD_SOURCE_SEED_KEY")
    source_seed.seed_status()

if __name__ == "__main__":
    main()
```