recursive_validation_engine.py

```python
from curriculum_engine import CurriculumEngine
from symbolic_overlay_engine import SymbolicOverlay
from identity_stack_engine import DiagnosticsEngine, IdentityStack

class RecursiveValidationEngine:
    def __init__(self):
        self.curriculum_engine = CurriculumEngine()
        self.symbolic_overlay = SymbolicOverlay()
        self.diagnostics_engine = DiagnosticsEngine()

    def build_full_system(self):
        self.curriculum_engine.build_base_structure()

        self.symbolic_overlay.add_symbol(
            name="Caduceus",
            axes="Vertical + Dual Helix",
            recursion_formula="Phi Spin x Dual Axis",
            notes="DNA mirrored recursion"
        )
        self.symbolic_overlay.add_symbol(
            name="Celestial Clock",
            axes="Orbital Rings",
            recursion_formula="Nested Orbital Resonance",
            notes="Cosmic timing recursion"
        )

        identity_a = IdentityStack(
            name="Subject Infinity",
            rna_memory="AACTGACTGACGTTG",
            dna_code="ATCGTTAGCTAGCTAG",
            xenial_state="Active Harmonic Sync"
        )

        self.diagnostics_engine.add_identity(identity_a)

    def run_validation(self):
        print("=== Curriculum Lattice ===")
        self.curriculum_engine.display_curriculum()
        print("=== Symbolic Overlay ===")
        self.symbolic_overlay.display_symbols()
        print("=== Identity Diagnostics ===")
        self.diagnostics_engine.run_diagnostics()

def main():
    engine = RecursiveValidationEngine()
    engine.build_full_system()
```

```
    engine.run_validation()

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