CPSC326 | Homework 4: Semantic Checker | 3/19/2025 | Arjuna Herbst

For the semantic checker, I created a series of tests to ensure that the areas I struggled with the most while completing the assignment were working properly, not just passing the provided tests. For my positive tests, I created a test to ensure that my semantic checker could handle struct creation and basic operations since I had some problems with them when I was implementing my visit methods.

As for my negative test cases, the first test I created tested my semantic checker to ensure that it could properly identify type mismatches when assigning arrays of one struct type to another. I struggled when checking for correct array accesses inside structs, so I created another test that looks for that.

Positive test cases inputs:

```
struct Person {
  name: string,
  age: int
}
void printPerson(p: Person) {
  if p != null {
    print(p.name)
    print(p.age)
  }
}
void main() {
  var p1: Person = new Person("Alice", 25)
  var p2: Person = new Person("Bob", 30)
  printPerson(p1)
  # Simple assignment operations
  p1.age = 26
  var name: string = p2.name
  # Null check
  if p1 != null {
    print("Person exists")
  }
}
```

```
int addOne(x: int) {
          return x + 1
        }
        int multiplyByTwo(x: int) {
          return x * 2
        }
        void main() {
          var x: int = 5
          # Basic function call chain
          var result: int = multiplyByTwo(addOne(x))
          # Use the result
          if result > 10 {
            println("Result is greater than 10")
          } else {
            println("Result is less than or equal to 10")
          }
        }
```

Positive test cases inputs:

```
        struct Point {x: int, y: int}
        struct Line {p1: Point, p2: Point}

        void main() {
          var points: [Point] = new Point[3]
          var lines: [Line] = points  # This should fail - type mismatch
        }
```

```
 struct Employee {name: string, id: int}

        void main() {
          var emp: Employee = new Employee("Alice", 101)
          print(emp.salary)
        }
```

```
void main() {
          var x: string = "Hello"
          var y: int = 5
          var z: string = x - y
        }
```