Lean Structures Exercises

Greyson Knippel
CPSC 326-01 — Gonzaga University, compiled with LaTeX
Professor Johnson

January 31, 2026

problem 1

part (a):

```
structure RectangularPrism where
  height : Float
  width : Float
  depth : Float
```

Part (b):

```
def volume (r1 : RectangularPrism) : Float :=
  r1.height * r1.width * r1.depth

def myPrism : RectangularPrism :=
  {height := 5.0, width := 2.0, depth := 6.0}

#eval volume myPrism
```

Part (c):

```
structure Segment where
  s1 : Float
  s2 : Float
  f1 : Float
  f2 : Float

def mySegment : Segment :=
  {s1 := 0.0, s2 := 0.0, f1 := 5.0, f2 := 4.0}

def length (m : Segment) : Float :=
  Float.sqrt (((m.f1 - m.s1)^2) + ((m.f2 - m.s2)^2))

#eval length mySegment
```

problem 2:

1. **strict:** functions and parameters are fully evaluated before the function body begins evaluation.
2. **pure:** programs cannot have side effects.
3. **functional:** functions are primary objects of interest; computation is evaluating functions (mathematical expressions).
4. **has dependent types:** types can contain programs that compute types.

problem 3:

```
structure Point3D where
  x : Float
  y : Float
  z : Float
```

problem 4:

```
def minimumComponent (p : Point3D) : Float :=
  if (p.x < p.y) then
    if p.x < p.z then p.x else p.z
  else
    if p.y < p.z then p.y else p.z
```

problem 5:

```
def midpoint (p1 : Point3D) (p2 : Point3D) : Point3D :=
  { x := (p1.x - p2.x) /2.0,
    y := (p1.y - p2.y) /2.0,
    z := (p1.z - p2.z) /2.0 }
```