CNC 2012
Programming Languages
Professor Sheryl Shulman
Homework chapter 5
Owen Meyer

5.2)
The Haskell Prelude provides a Boolean type that is ordered.
data Bool = False | True
This gives the ordering False < True
A function could be written that converts a Bool to another type which might be usefull for something like converting bits as ints to booleans to use existing boolean functions.

5.3)
Keeping types and their associated operators from becoming muddled up with other types in confusing ways makes for sensible strongly typed programming.

5.5)
.2 → 0 01111100 10011001100110011001101
.5 → 0 01111110 00000000000000000000000
.3 → 0 01111101 00110011001100110011010
1. → 0 01111111 00000000000000000000000

5.7)
```
union intfloat {
        int i;
        float f;
}
intprint(read(intfloat.i));
floatprint(intfloat.f);
```

5.9)
Big-endian has the bytes numbered from the most-significant-bit to the least-significant-bit.
Little-endian has the bytes numbered from the LSB to the MSB.

5.14)
First, maybe the desired result is to have the var a point to the array referenced by b: a = *b
Or, maybe the desired result is to have the var a contain the value of b's address: a = &b

5.19) [(or {(means meta [or {
Concrete Syntax:

Declaration → Type Identifier [([Integer])] {(,Identifier [([Integer])])} ; |
struct identifier { Declarations }

Assignment → Identifier [([Expression])] = Expression; | Identifier . Identifier = Expression

Expression → …
…

Primary → Identifier [([Expression])] | Literal | (Expression) | Type (Expression) |
Identifier . Identifier

Abstract Syntax:

Declaration → VariableDecl | ArrayDecl | StructDecl
…
StructDecl → Declarations

Assignment → VariableRef target; Expression source;
VariableRef → Variable | ArrayRef | StructRef
StructRef → String id; VariableRef field;

Expression → <no change as VariableRef handles StructRefs>