Homework 8

Problem 1

a)

A — no supertypes

B — supertype A

C — supertype A

D — supertypes A, B, C

E — supertypes A, C

F — supertypes A, B, C, D

G — supertypes A, B

b)

Any class that inherits from interface B could have their objects passed into foo — this includes D, F, G as shown in (a)

c)

You can pass a subtype into a function that takes its supertype, but since A is the supertype to C, you cannot pass an instance of A into C. Therefore, the function bletch cannot call bar.

Problem 2

The goal of inheritance is to reuse and extend code with an “is-a” relationship.

The goal of subtype polymorphism is to refer to an object to which its subtype objects can pass their functions into and receive expected behavior.

The goal of dynamic dispatch is to call a version of an overridden method during runtime instead of compile time based on the type of object that calls the method.

The difference between these three is that inheritance focuses on reusing/extending code, polymorphism creates a common interface for different, but similarly based, data types, and dynamic dispatch is about determining which method implementation to utilize at runtime.

Problem 3

Subtype polymorphism can be utilized similarly with duck typing in a dynamically typed language, but we don’t because subtype polymorphism typically enforces typing at compile time for a statically typed, while dynamic typed languages have types evaluated at runtime. Since a variable in a dynamic language can be assigned different types and values at different points of a program, the traditional concept of subtype polymorphism is less applicable unless you intentionally modify a variable to have the behavior necessary to duck type.