Assignment 2 - Question 1

In a purely object-oriented language, the need for a traditional switch construct, like in C++, is reduced due to the combined effects of polymorphism, dynamic typing, and language simplicity. Polymorphism allows objects to respond to method calls in a way specific to their type. Therefore, instead of using a switch statement to determine an object's type and act on it, you can instead directly call a method that each object type knows how to handle. Dynamic typing further supports this by determining method calls at runtime based on the object's actual type. This allows the same code to work with different types of objects without needing a switch to distinguish between them. Finally, the simplicity of purely object-oriented languages often encourages designs that encapsulate behavior within objects, reducing the need for control structures like switch and leading to more modular and maintainable code.