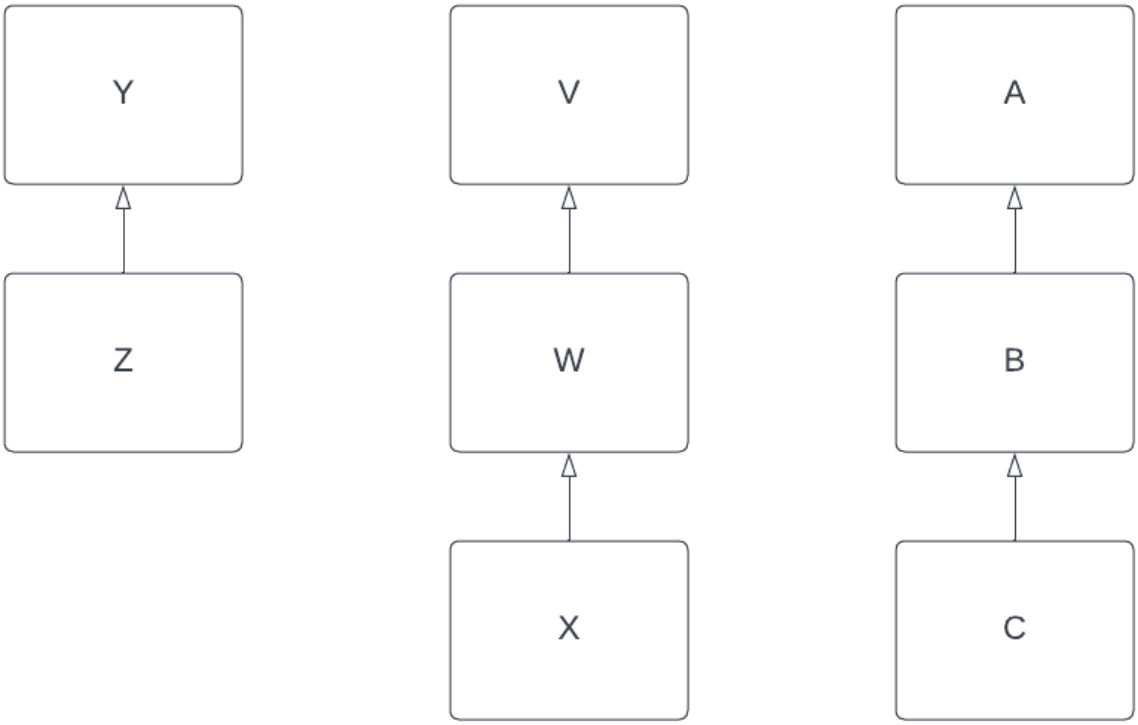
Chapter 16: #3

- c1 = c2, Allowed in java since C2 is a derived class of C1.
- c1 = c3, Doesn't allow since C1 and C3 are different classes and don't relate to one another.
- c1 = c4, Doesn't allow since C1 and C4 are different classes and don't relate to one another
- c1 = i1, Allowed with a downcast since C1 implements I1
- c1 = i2, Allowed with a downcast but may cause an exception at runtime
- c2 = c1, Allowed with a downcast since C2 is a derived class of C1
- c2 = c3, Doesn't allow since c2 and c3 are different classes, and java doesn't allow downcasts on interfaces.
- c2 = c4, Doesn't allow since c2 and c4 are different classes, and java doesn't allow downcasts on interfaces.
- c2 = i1, Allowed with a downcast since C2 inherits from C1, and C2 is implementing I1
- c2 = i2, Allowed with a downcast since C2 implements I2 but may cause an exception at runtime.
- c3 = c1, Doesn't allow since c3 and c1 are different classes, and java doesn't allow downcasts on interfaces.
- c3 = c2, Doesn't allow since c3 and c1 are different classes, and java doesn't allow downcasts on interfaces.
- c3 = c4, Allowed since C4 is a derived class of C3.
- c3 = i1, Allowed with a downcast since C3 implements I1
- c3 = i2, Allowed with a downcast but may cause an exception at runtime
- c4 = c1, Doesn't allow since c4 and c1 are different classes, and java doesn't allow downcasts on interfaces.
- c4 = c2, Doesn't allow since c4 and c2 are different classes, and java doesn't allow downcasts on interfaces.
- c4 = c3, Allowed with a downcast since C4 is a derived class of C3
- c4 = i1, Allowed with a downcast but may cause an exception at runtime
- c4 = i2, Allowed with a downcast since C4 implements I2 but may cause an exception at runtime.
- i1 = c1, Allowed since C1 implements I1 and a C1 object has all the stuff for i1
- i1 = c2, Allowed since C2 indirectly implements I1 through C1 and a C2 object has all the stuff for i1
- i1 = c3, Allowed since C3 implements I1 and a C3 object has all the stuff for i1
- i1 = c4, Allowed since C4 indirectly implements I1 through C3 and a C4 object has all the stuff for i1
- i1 = i2, Allowed with a downcast but may cause an exception at runtime.
- i2 = c1, Allowed with a downcast but may cause an exception at runtime.
- i2 = c2, Allowed since C2 implements I2 and a C2 object has all the stuff for i2
- i2 = c3, Allowed with a downcast but may cause an exception at runtime
- i2 = c4, Allowed since C4 implements I2 and a C4 object has all the stuff for i2
- i2 = i1, Allowed with a downcast but may cause an exception at runtime

	C1	C2	C3	C4	I1	I2
C1	trivial	yes	no	no	cast	cast
C2	cast	trivial	no	no	cast	cast
C3	no	no	trivial	yes	cast	cast
C4	no	no	cast	trivial	cast	cast
I1	yes	yes	yes	yes	trivial	cast
I2	cast/yes	cast	yes	cast	trivial	trivial

Part A - Ordered List/possible combinations and diagram
Note: I used Repl.it to run my main.cpp file
Diagram:



Possible Combinations:
Y V A *
Y V B
Y V C
Y W A
Y W B *
Y W C

Y X A
Y X B
Y X C *
Z V A *
Z V B
Z V C *
Z W A
Z W B
Z W C
Z X A *
Z X B *
Z X C