3

C1 to C2: Disallowed because a superclass reference cannot point to a subclass object without a cast.   
C1 to C3: Disallowed because they are unrelated classes.  
C1 to C4: Disallowed because C1 and C4 are unrelated classes.  
C1 to I1: Allowed because C1 implements I1.  
C1 to I2: Disallowed because C1 does not implement I2.  
C2 to C1: Allowed because C2 is a subclass of C1.  
C2 to C3: Disallowed because they are unrelated classes.  
C2 to C4: Disallowed because they are unrelated classes.  
C2 to I1: Allowed because C2 (through C1) implements I1.  
C2 to I2: Allowed because C2 implements I2.  
C3 to C1: Disallowed because they are unrelated classes.  
C3 to C2: Disallowed because C2 is not a superclass of C3.  
C3 to C4: Disallowed without a cast because C4 is a subclass of C3. With a cast, it is allowed.  
C3 to I1: Allowed because C3 implements I1.  
C3 to I2: Disallowed because C3 does not implement I2.  
C4 to C1: Disallowed because C1 is not a superclass of C4.  
C4 to C2: Disallowed because C2 is not a superclass of C4.  
C4 to C3: Allowed because C4 is a subclass of C3.  
C4 to I1: Allowed because C4 (through C3) implements I1.  
C4 to I2: Disallowed because C4 does not implement I2.  
I1 to C1: Allowed with a cast since an interface can be cast to any of its implementing classes.  
I1 to C2: Allowed with a cast because C2 is an implementation of C1, which implements I1.  
I1 to C3: Allowed with a cast since C3 implements I1.  
I1 to C4: Allowed with a cast because C4 is an extension of C3, which implements I1.  
**I1 to I2: Disallowed because they are unrelated interfaces.  
I2 to C1: Allowed with a cast since it's possible for an I2 reference to refer to a C1 object, provided that object is actually an instance of C2, which implements I2.  
I2 to C2: Allowed because C2 implements I2.  
I2 to C3: Disallowed because C3 does not implement I2.  
I2 to C4: Disallowed because C4 does not implement I2.  
I2 to I1: Disallowed because they are unrelated interfaces.**

A

C1 = C2 (Direct Assignment Allowed): A variable of type C1 can directly reference an instance of C2 because C2 is a subclass of C1, hence they share the same interface.

C2 = C1 (Cast Required): To assign a C1 instance to a C2 variable, a cast is required because C2 may have additional members that C1 doesn't, and we need to ensure that the object actually is an instance of C2.

C1 and C3 (Incompatible): No assignment is possible between C1 and C3 as they are unrelated classes with no inheritance relationship.

C1 to I1 (Direct Assignment Allowed): An instance of C1 can be assigned to an I1 variable without a cast because C1 implements the I1 interface.

I1 = C1 (Direct Assignment Allowed): An I1 variable can reference an instance of C1 directly since C1 implements I1.

C2 to I1 (Cast Required): Assigning an instance of C2 to an I1 variable may require a cast because C2 inherits I1's interface through C1, but I1 does not have all of C2's specific members.

C3 and C1 (Incompatible): No assignment is possible between C3 and C1 because they don't have a direct or indirect inheritance relationship.

C4 to C3 (Direct Assignment Allowed): A C3 variable can directly reference an instance of C4 because C4 is a subclass of C3.

C4 to I1 (Direct Assignment Allowed): A variable of type I1 can reference an instance of

C4 directly since C4, through its superclass C3, implements I1.

I2 and C4 (Incompatible): No direct assignment is possible between I2 and C4 because there is no inheritance relationship between them.