1. **What is the difference between “downcasting” and “upcasting”?**

   Upcasting is referring to an object as if it were a superclass. Downcasting is converting an object back to a subclass.

2. **Why would you mark a method as virtual? Why wouldn’t you?**

   You mark a method virtual when you want a subclass to be able to override that method. You would not mark it virtual when you 100% do not need it because it can make the code slower.

3. **What is the difference between a dynamic cast and a static cast? Why would you use one over the other?**

   A dynamic cast looks at the actual type of object that is being cast and checks if it is legal to cast it to the requested type. Statis cast checks only what it can when compiled, but does not look at the actual type of object at runtime. Dynamic casting is slow because it requires searching through the entire inheritance tree, so if you really needed to make the code faster you would use a static cast.

4. **When might you use a const cast?**

   You might use a const cast when you are using a library for which you do not control the code. The writer of that code may have forgotten to mark a certain method const, and if you have a const object in your own code you would be unable to use that method.

5. **Suppose you have a Fruit base class and a Banana subclass. Suppose you have a banana declared like this:**
   Banana b;
   What is the difference between what happens when you pass your banana to a method de- clared like this:
   void eat(Fruit f);
   vs. one declared like this:
   void eat(const Fruit& f);.

   When you pass banana to the first method, it can alter the Banana object. When you pass banana to the second method, it can not alter the banana object because there is a const.

5. **What is the difference between how Java and C++ handle multiple inheritance?**

   In every java class, except for object, there is only one parent class. In C++, a class can have multiple parent classes or even no parent classes. In java, everything inherits from the object class. In java you have interfaces which group classes different than the normal inheritance tree. In C++ you can inherit from more than one superclass and access the methods and data implementations that superclass has.