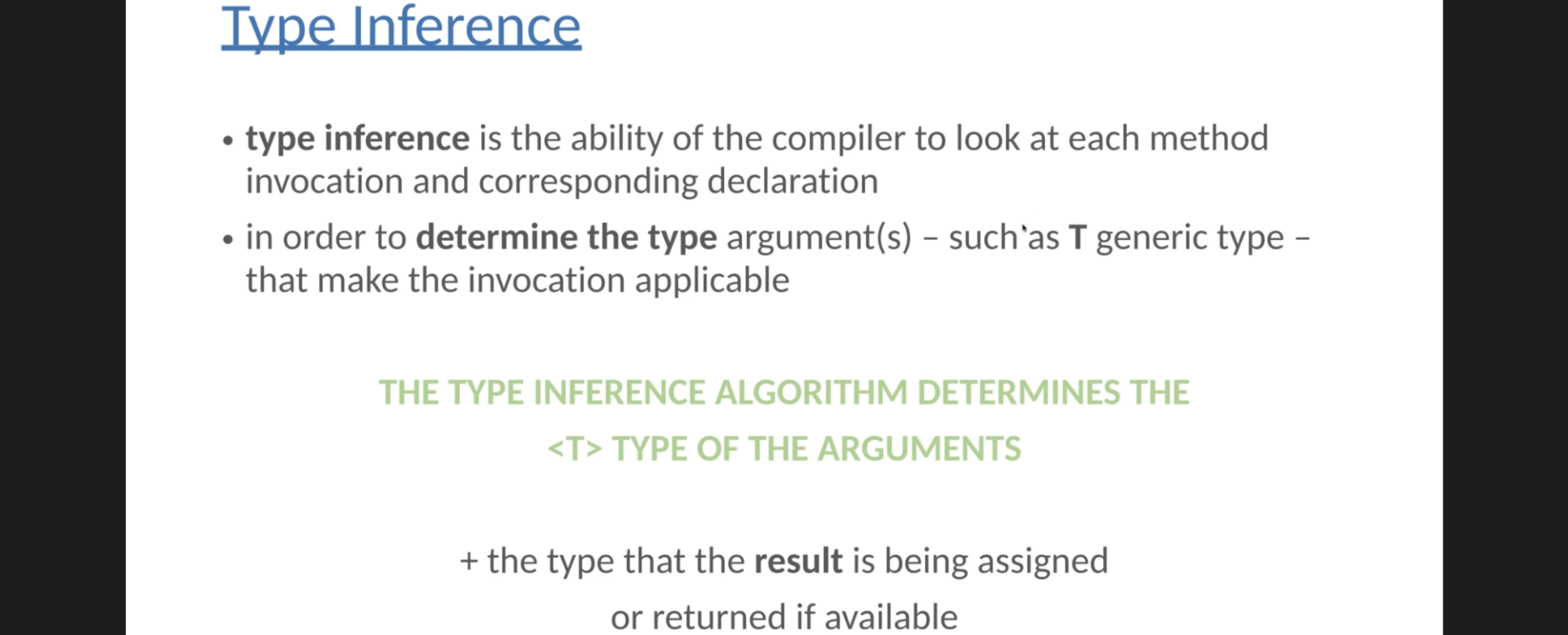
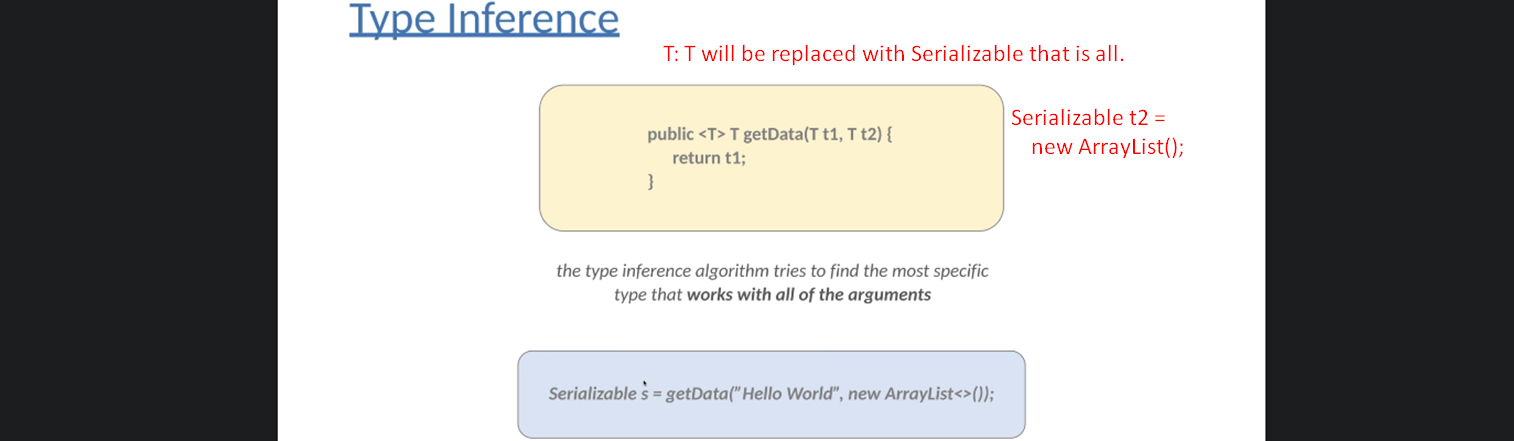
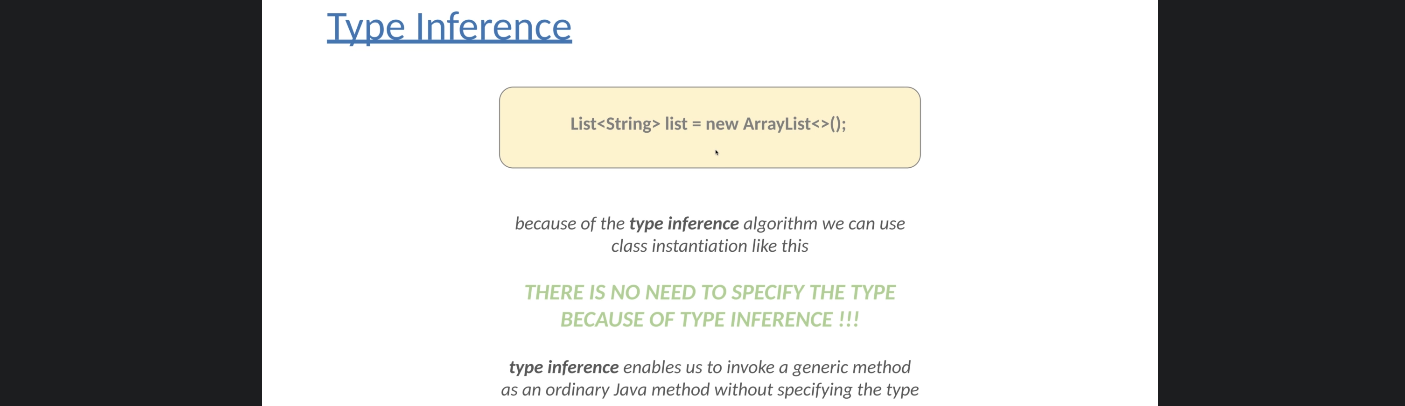
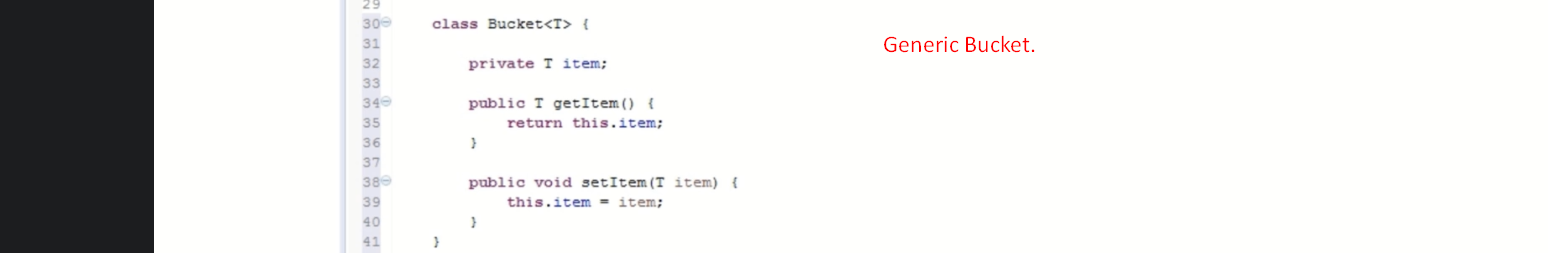
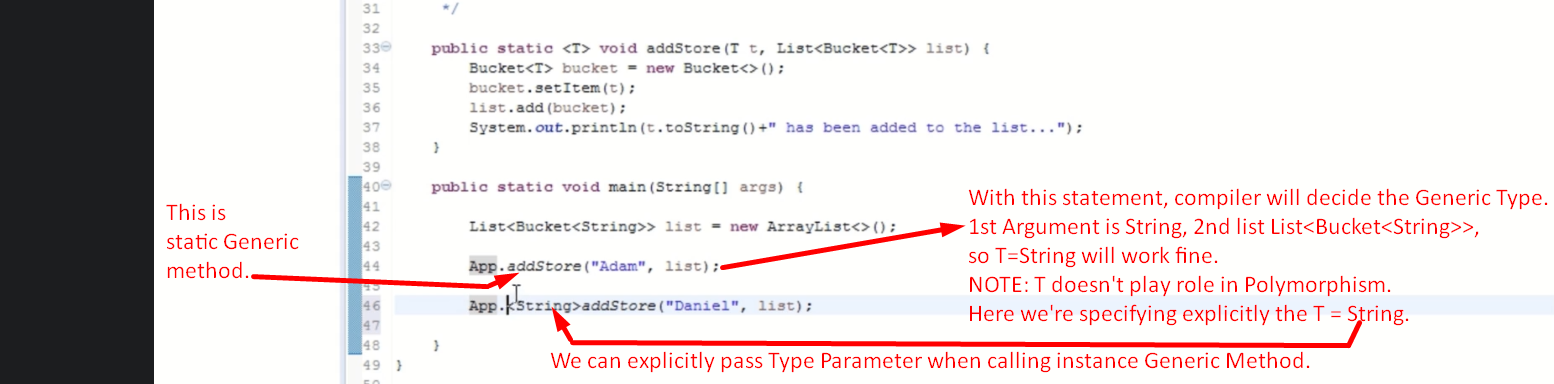
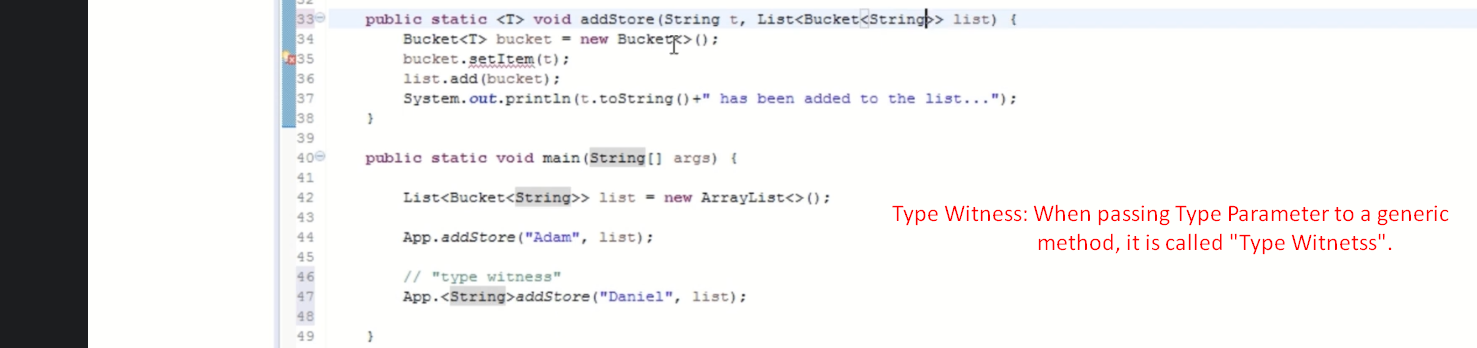
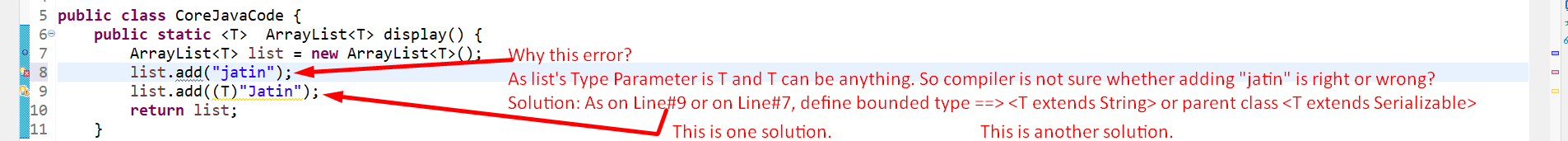
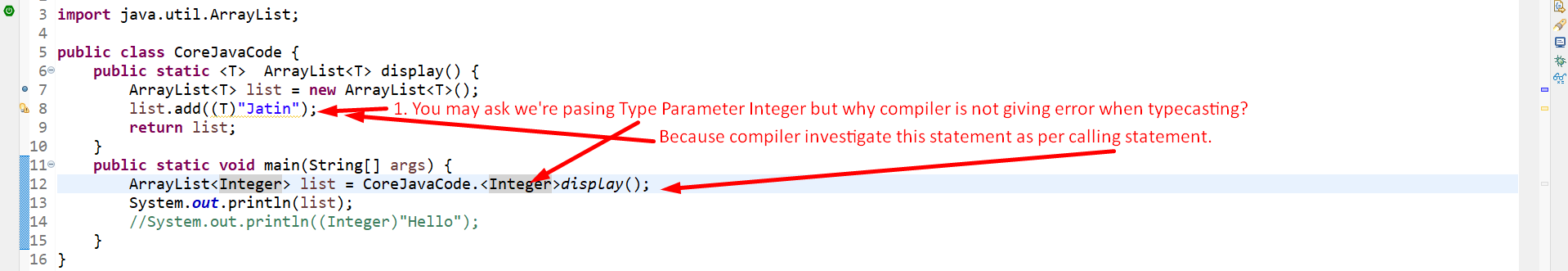
1. 
2. 
3. 
   1. In the above example, we don’t need to typecast the retuned value as under the hood, the compiler is going to find the most specific type that works with all of the arguments.
   2. Actually, see the above example carefully. We don’t want to use typecast when storing the returned value in variable Serializable s.  
      So, compiler will check if Serializable can be Generic Type.   
      Yes as both passed arguments are also Serializable (“Hello World”, new ArrayList<>()).   
      So, compiler decides Generic Type to be Serializable.
4. **Another Feature of Type Inference:**
   1. ****
5. **Let’s see an example:**

****

****

1. Type Witness  
   
2. d
3. Jatin 🡺 Let’s consider some cases:  
    **Correction**: As compiler doesn’t consider the calling statement at all.   
    