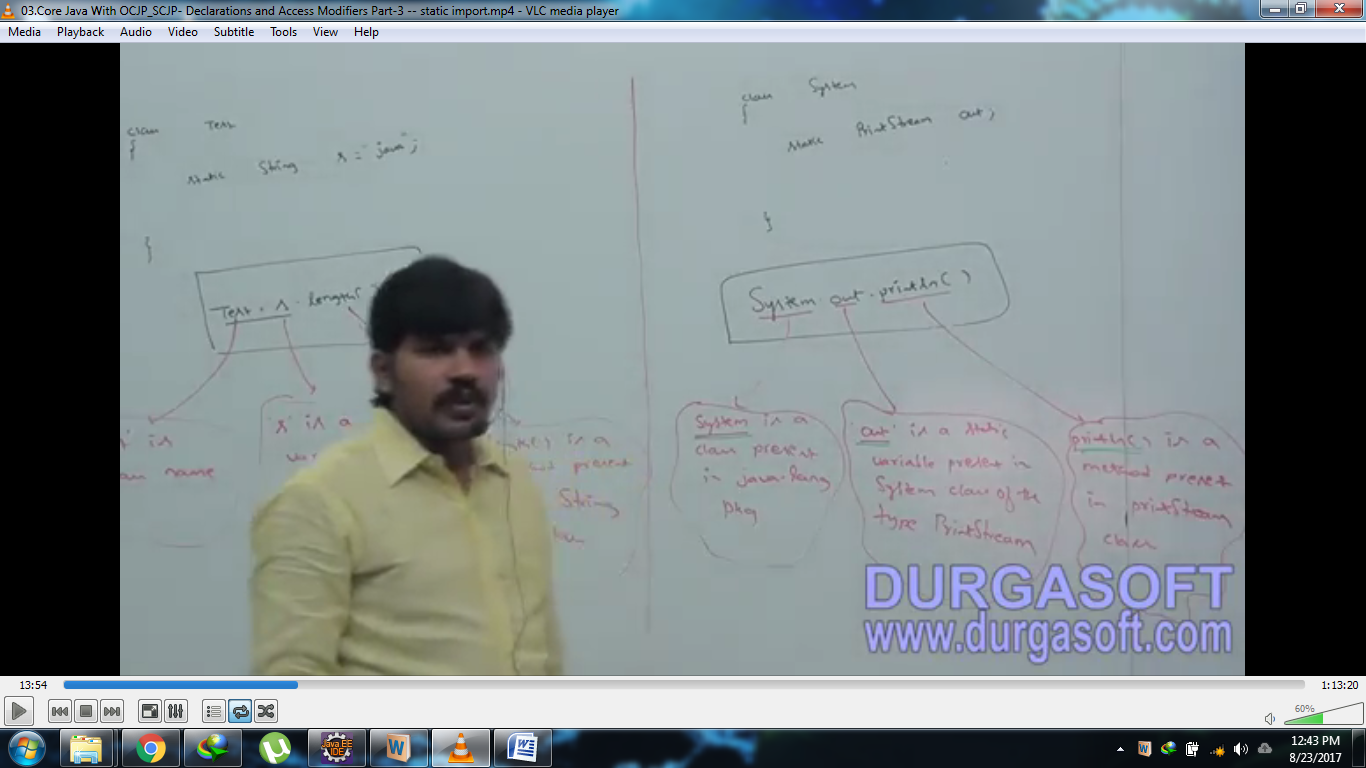
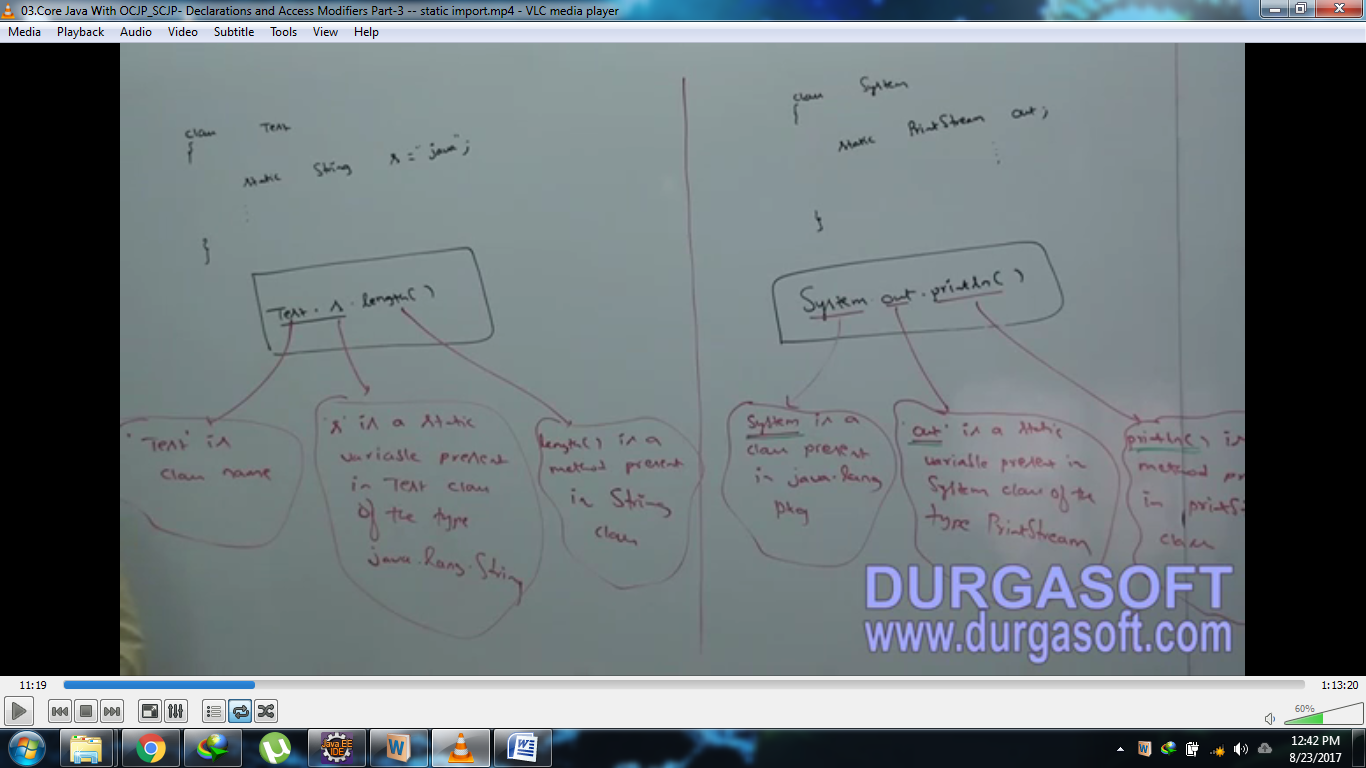
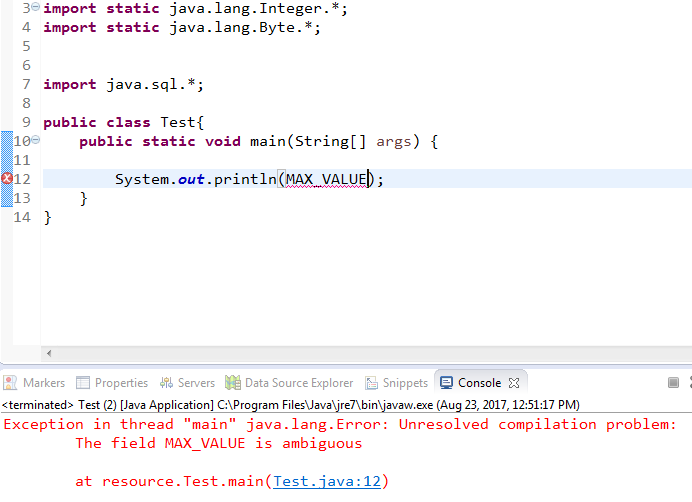
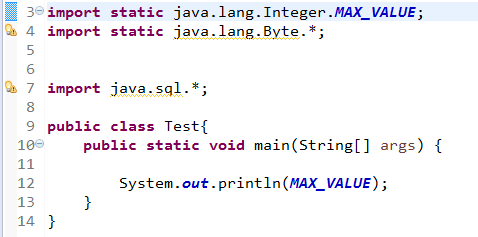
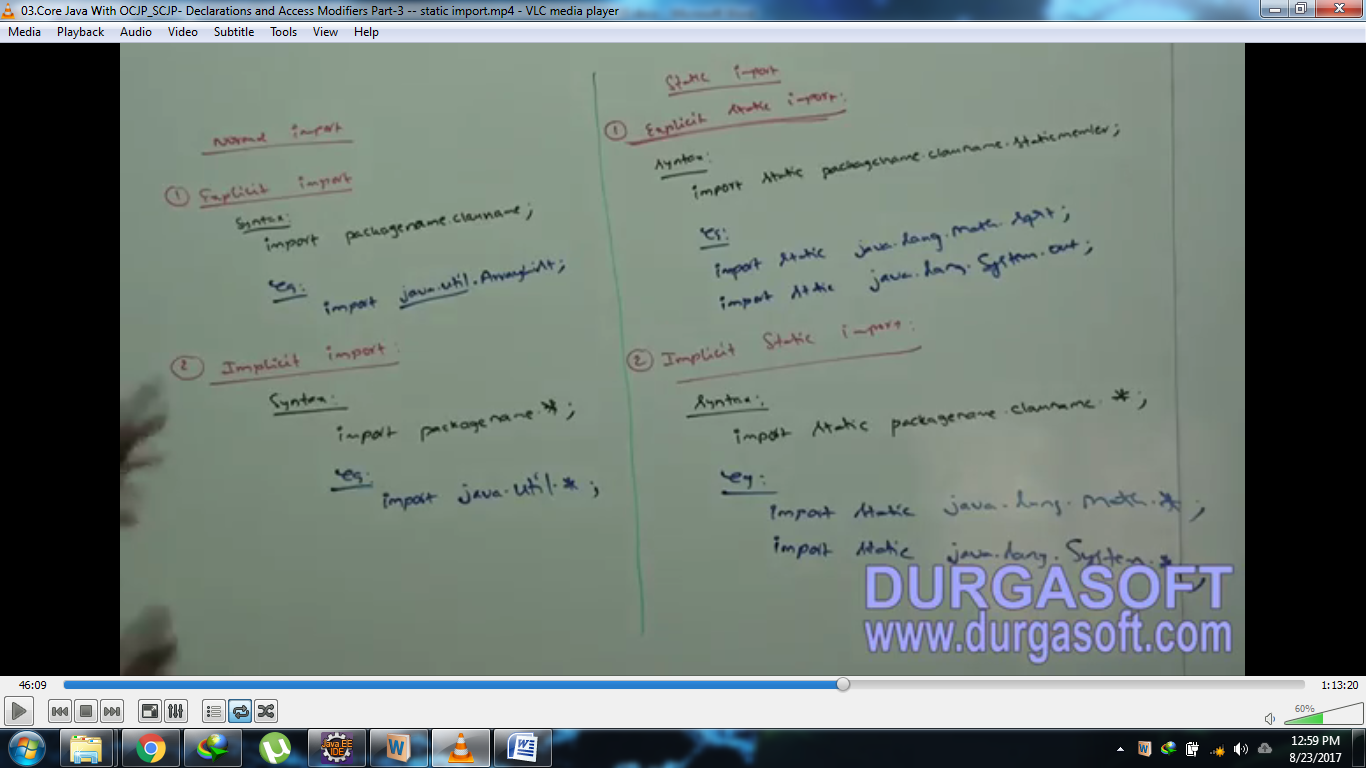
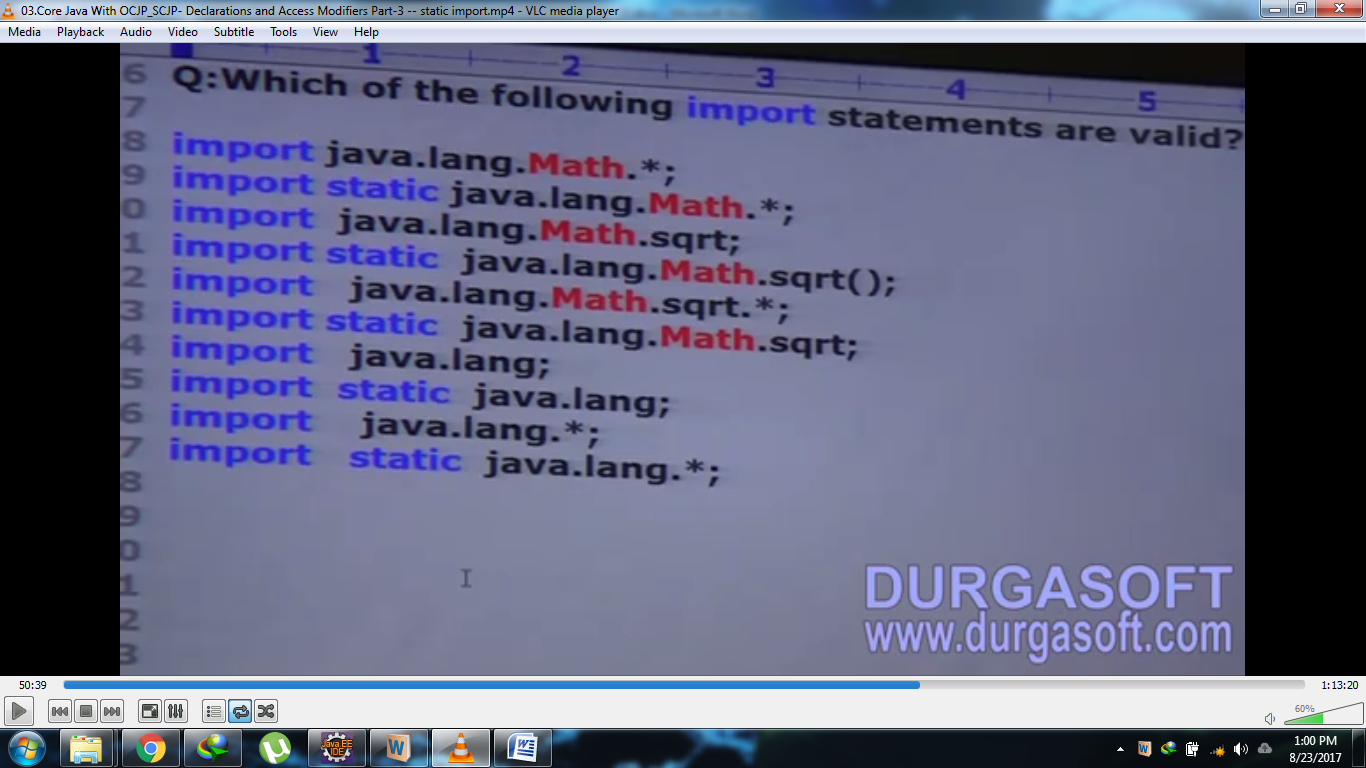
1. **Explain about System.out.println():**
   1. ****out is a static variable in System class hence we can access by using class named System. But whenever we are using static import, it’s not required to use class name and we can access directly like out.println(“hello”)
2. **  
   Solution: Explicit static import  
   **

Explicit Static import

both Integer and Byte wrapper classes contain  
MAX\_VALUE 🡪 Ambiguous

1. **Order of static variable resolution.**
   1. Current Static member
   2. Explicit Static import
   3. Implicit static import.
2. **Normal Import and Static Import  
   **
3. ****
4. **NOTE**: Two packages contain a class or an interface with the same name is very rare. Hence, ambiguity problem is very normal import. But two classes or interfaces may contain a variable or method with same name which is very common and hence, ambiguity problem is also very common in static import.
5. **Difference b/w normal import and static import**?
   1. **Normal import**: To import a class or an interface of a particular package to avoid fully-qualified name so that we can use short name.
   2. **Static import**: To import static members of a particular class or interface. In this case, it’s not required to use class name to access static membes.
6. d