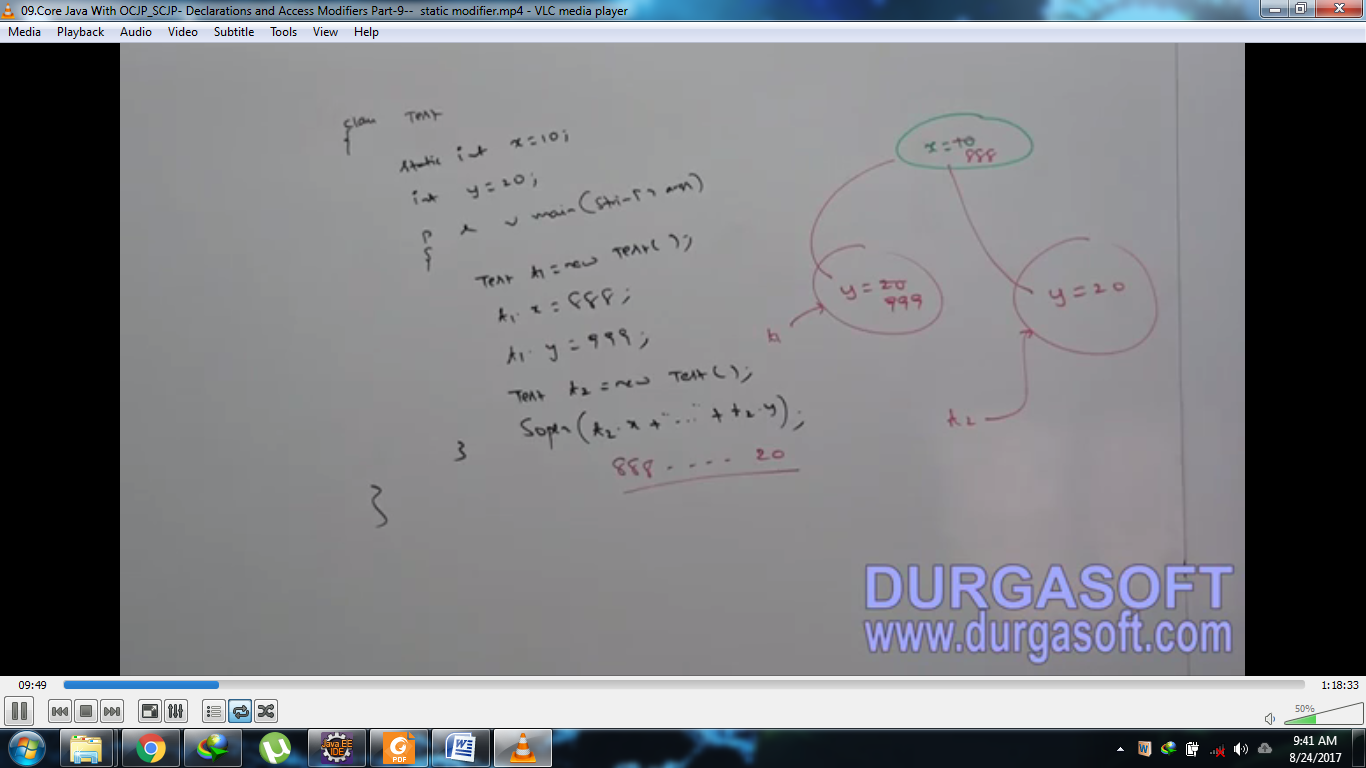
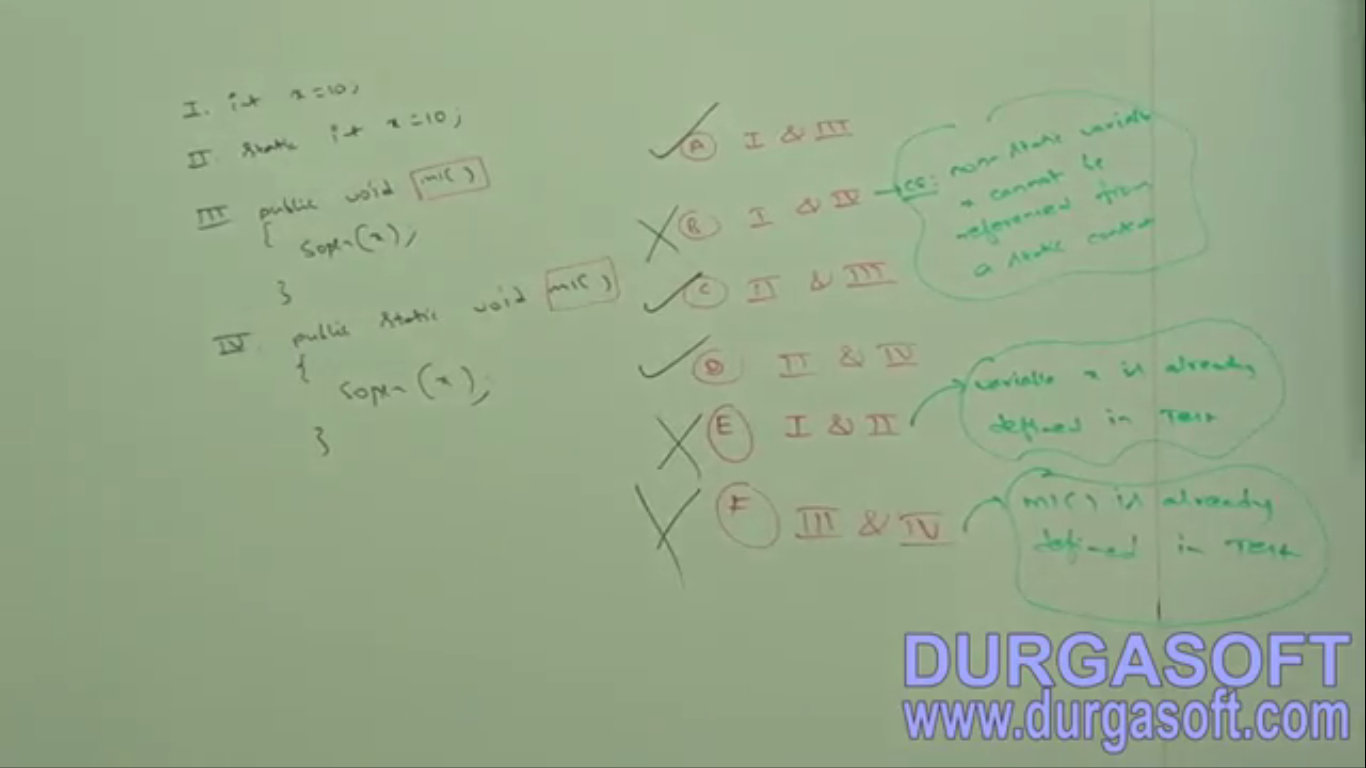
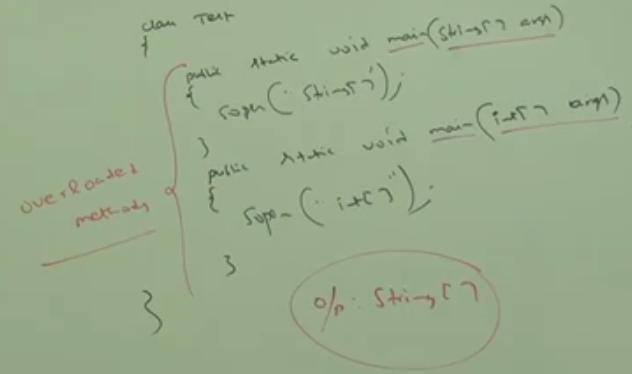
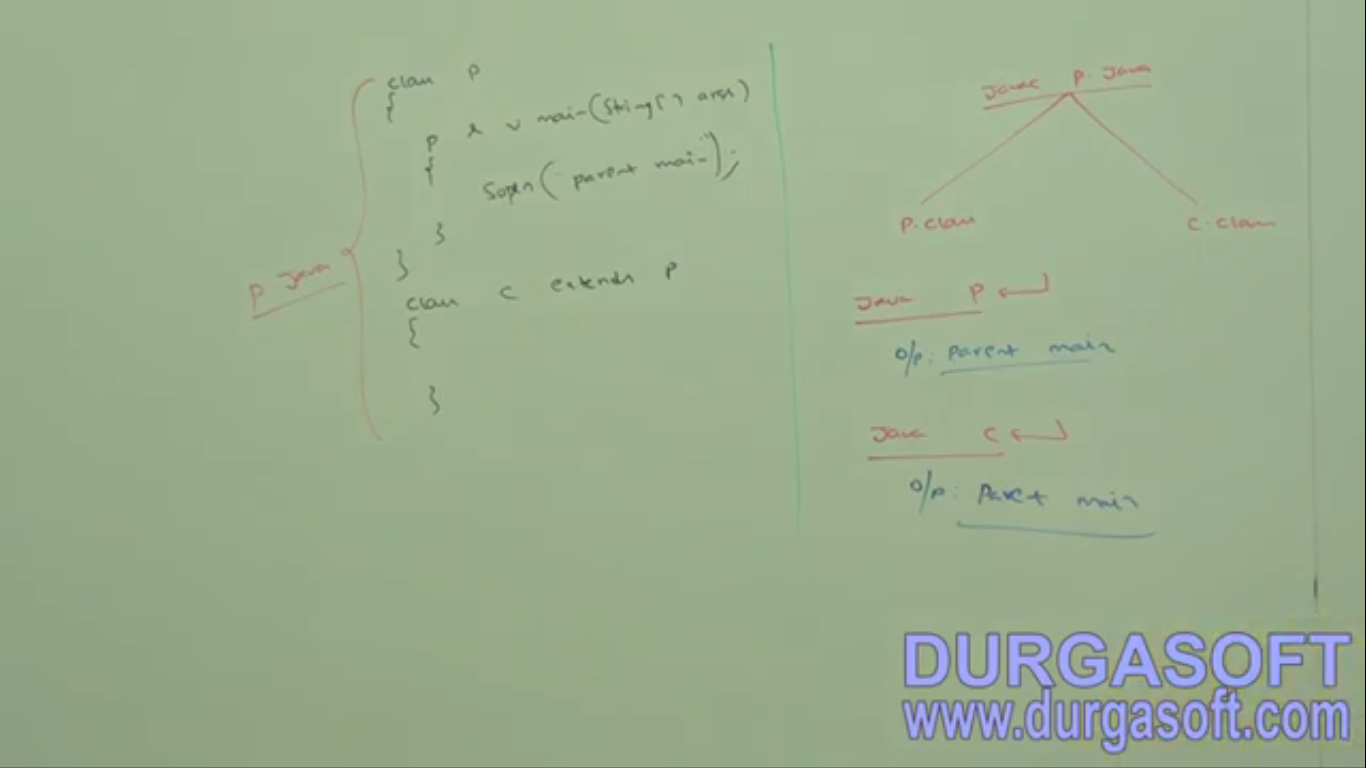
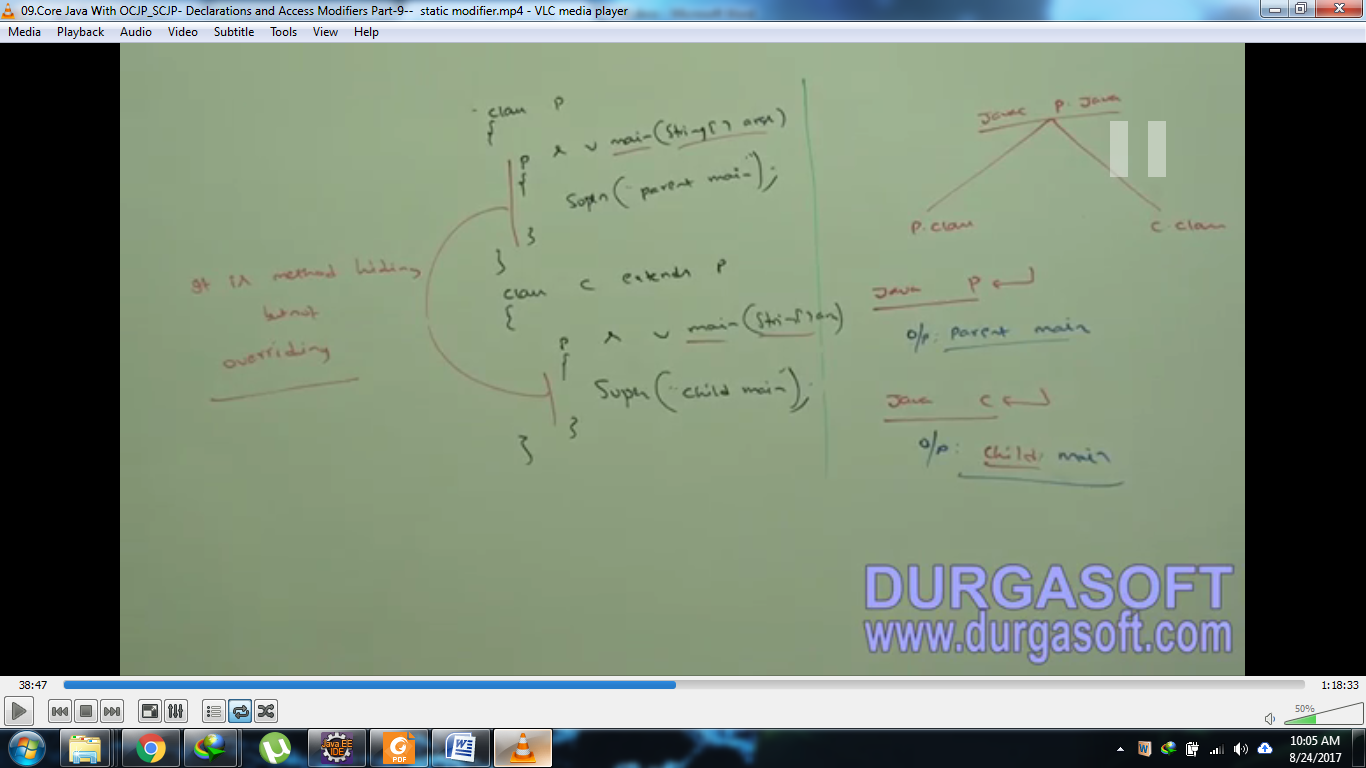
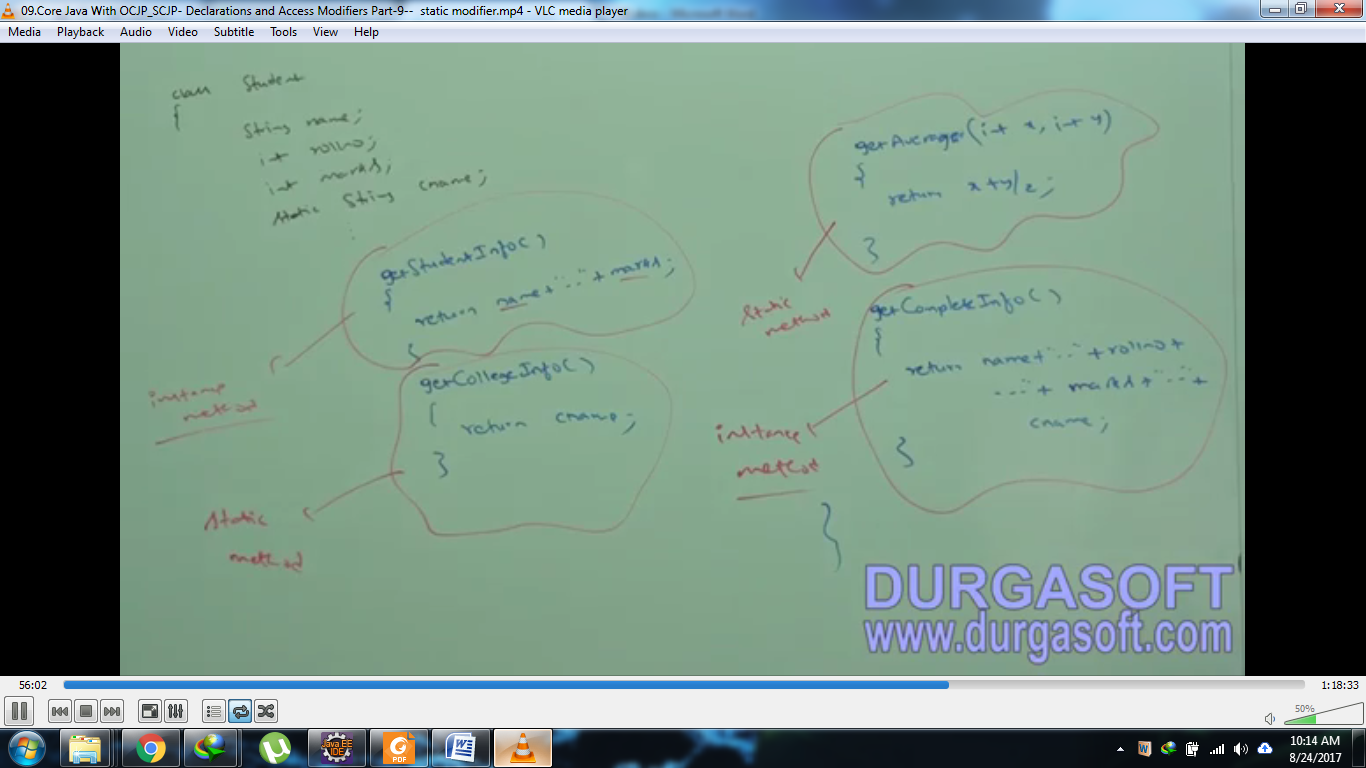
Static Modifier

1. Static is a modifier applicable for methods and variables but not for top-level classes.
2. We can’t declare top-level classes to be static but inner classes (such type of inner class is called **Static Nested Class**).
3. **Static variable**: In the case of instance variables, for every object, a separate copy will be created. But for static variables, a single copy will be created at class level and shared by every object of that class.   
   
4. We can’t instance members directly from static area but we can access from instance directly.   
   We can access static members from both instance and static areas directly.
5. Consider the following declarations:  
   

Which combinations are valid  
NOTE: static variables are accessible directly in both instance method and static method. ☺

1. **Let’s study some cases about static methods.**
   1. **case 01**: Overloading concept is applicable for static methods including main() but JVM always call main(String… args) only.  
      
   2. **Case 02**: Inheritance concept is applicable for static methods including main(String… args). Hence, while executing child class, if child doesn’t contain main(String… args), then parent class main(String… args) will be executed.  
      
   3. **Case 01**: if both parent and child classes define their own main(String… args) then   
      
2. For static methods, Overloading, inheritance concepts are applicable but overriding concept is **not applicable** but instead of overriding, method hiding concept is applicable.
3. Which methods should be declared to be static and which to be non-static?
   1. static: if there is no reference to instance variable then static.
   2. Non-Static: If there is even single reference to instance variable to instance variable then Non-Static.
   3. **Example**:



1. For static method, implementation should be available whereas for abstract method implementation is not available. Hence, abstract static combination is not allowed (illegal) for methods.

Synchronized

1. Synchronized is the modifier applicable for where executable code can be written   
   Such as 🡪 Methods, Blocks  
   Not applicable for classes and variables.
2. If multiple threads are trying to operate simultaneously on the same java object, then there may be chance of **“Data Inconsistency Problem”** this is called **“Race Condition”**. We can overcome this problem by using **synchronized keyword**. If a method or block is declared as synchronized then at a time only one thread is allowed to operate/execute that method or block on the given object so that Data Inconsistency Problem is resolved.

**Disadvantage**:   
It increases the waiting of threads and creates performance problem. Hence, if there is no specific requirement then it’s not recommended to use synchronized keyword.   


1. Synchronized method should have implementation as synchronized keyword says that this method implementation should executed by one thread at a time.
2. NOTE: abstract synchronized combination is not illegal as synchronized talks about implementation whereas abstract method doesn’t have any implementation.