### **Some things that ignore**

* When a class throws and exception, check of all kinds of checked exceptions are caught or not.
  + If we throw IOException and in catch we only have FiileNotFound, it will result in compilations error.
* Definition like int = ‘5’ is valid. Int is compatible with char, but char[] is not compatible with int[]
* When we have if conditions always check if the variables are initialized.
* A sub-type cannot refer to an object of super type.
* StackOverflowError is used for resolving infinite Loops.
* If some line of code doesn’t throw any particular type of exception and we try to catch it, we will get compilation error
  + Eg: if we only have an sopln in main function and we try to catch a FIleNotFound for that line of code we will get compilation error.
* If we use equals on different kind of Objects, then we get FLASE.
* toString() methods doesn’t use String Literals.
* LocalDate is final class and cannot be extended.
* Strings are immutable.
* When iterating a list use iterator to remove, as we might get concurrent use exception.
* Import zxy.\*; export class in zxy and not the subdirectories.
* Int case we use integer in remove() as a parameter it will remove the index, and return the deleted element, else it removes the particular Object and return TRUE is case of successful deletion.
* Integer Wrapper class ins immutable.
* Primitive and Strings are executed in compile time.
  + String abc = ‘’ONE”; //complie time
  + String abc2 = s1 + s2; This is done and run time and it’s not from String pool.
* Valid Switch Datatypes
  + Primitive: int, byte, short, char
  + Wrapper: Integer, Byte, Short, Character, String, Enum.
* Boolean wrapper value by default is NULL.
* StringBuilder doesn’t override equals(), so reference is checked.
* If has a final variable, it may make the code unreachable to we get compilation error.
* List<int>is not valid.
* An interface method is implicitly public and abstract.
* isEmpty() on null gives null pointer.
* If we add elements to a list in a particular index, we can do so only till the length of the list, index above that will give compilation error.
* String is a final class.
* Local variables cannot be declared private, only modifier allowed is final.
* Correct Way to access static method of an interface is by using the name of the interface.
* Instance method of a subclass cannot override static method of SuperClass.
* Static methods can only access static members.
* Sub-Type can’t refer to an instance of super type so, at runtime
  + M obj = new M() ; // M is superclass of N  
    N obj2 = (N)obj1; // throws ClassCast Exception
* It is legal for constructors to have throws Clause.
* The child constructor should be of same type or a superclass in its Throws declaration.
* String Concatenation: If one of the Operand is String, then String conversion is performed on other expressions as well. NULL is converted to “null”. toString() methods is invoked for the objects.
* It is not possible to reach two levels up using super.super . (Only one level is possible).
* An overridden method cannot be declared to throw any Checked Expression. Only subtype. Though it can throw Runtime and Error.
* Static method of a subclass cannot hide the instance method of super class.
* Order of exception in multi-catch statement doesn’t matter.
* When variable is hidden check for reference. (Parent and child have the same variable name check the reference of the object.)
* Implicit Casting > var args
* Final boolean flag;  
  flag = false // doesn’t make the flag a complie time constant.
* Generics polymorphism is not allowed  
  ArrayList<Animal> list = new ArrayList<Dog>(); // not valid.
* Like any other method main can be overloaded, But main method called by JVM is always String[] parameter.
* For(;;) is an infinite loop.
* Java complier does implicit casting or wrapping but not both.
* When an abstract class inherits 2 interfaces with same method, we need to override the method  
  InterfaceName.super.profit();
* Interface Variable: public static final
* Interface Method: abstract and public.