**Java-8 – Anonymous Inner Class with Lambda Expression**

* **Part\_02:**

We can’t always replace anonymous inner class with Lambda expression.

Anonymous inner class that extends concreate class:

class Test{

}

Test test = new Test(){

};

Anonymous inner class that extends abstract class:

abstract class Test{

}

Test test = new Test(){

};

Anonymous inner class that implements an interface which contains multiple method.

interface Test{

public void m1();

public void m2();

public void m3();

}

Test test = new Test(){

public void m1(){}

public void m2(){}

public void m3(){}

};

Anonymous Inner class implements an interface which contains only one abstract method:

interface Test{

public void m1();

}

Test test = new Test(){

public void m1(){}

};

**Conclusion:**

Only the last case (anonymous inner class which implements an interface which contains only one method) can replaced with lambda expression, others are not possible.

Anonymous inner classes are more powerful than lambda expressions.

Anonymous Inner Class != Lambda Expression