**Java-8 – Method Reference By : : double colon operator – Part-01**

* **Interface implementation using lambda expression:**

interface Interf{

public void m1();

}

class Test{

public static void main(String[] args){

Interf I = () -> {

System.out.println(“Lambda expression”);

};

i.m1();

}

}

In case of lambda expression, method m1 is referenced through lambda expression.

There is another option to refer the method of an interface, that is using double : operator.

* **Double :: Operator:**

interface Interf{

public void m1();

}

class Test{

public static void m2(){

System.out.println(“Method reference”);

}

public static void main(String[] args){

Interf I = Test :: m2;

i.m1();

}

}

In above code, whenever we call Interf.m1(), it will refer the method m2(). This feature is called method reference.

The biggest advantage of method reference is code reusability.

Note:

To use method reference both methods should have same argument type.

Return types can be different, modifiers can be different and method can be of any type static or instance method.

**Important Note:**

Lambda expression can be replaced with method reference.

* **Example:**

interface Interf{

public void m1();

}

public class Test{

public static void m2(){

System.out.println(“Implementation by method reference”);

}

public static void main(String[] args){

Interf I = Test :: m2;

i.m1();

}

}