**Java-8 – Unary Operators and its primitive**

* **Agenda:**

**UnaryOperator**

**BinaryOperator**

* **UnaryOperator:**

interface Function<T, R>{

pubic R apply(T t);

}

Function<Integer, Integer> f = i -> i\*i;

System.out.println(f.apply(10));

In the above example both the input and output are same type Integer. Then we can make use of UnaryOperator.

UnaryOperator is the child class of Function.

interface UnaryOperator<T>{

public T apply(T t);

}

UnaryOperator<Integer> o = i -> i \* i;

System.out.println(o.apply(10));

Example:

class Test{

public static void main(String[] args){

UnaryOperator<Integer> f = i-> i\*i;

System.out.println(f.apply(10));

}

}

* **Primitive Unary Operator:**

interface IntUnaryOperator{

public int applyAsInt();

}

Example:

class Test{

public static void main(String[] args){

IntUnaryOperator f = i->i\*i;

System.out.println(f.apply(10));

System.out.println(f.apply(6));

}

}

interface LongUnaryOperator{

public long applyAsLong();

}

interface DoubleUnaryOperator{

public double applyAsDouble();

}