LAMBDA EXPRESSIONS: FUNCTIONAL CONSTRUCTS IN AN OBJECT ORIENTED CONTEXT - II

LAMBDA FUNCTIONS ARE ESPECIALLY POWERFUL WHEN THE OUTPUT OF ONE IS FED AS THE INPUT INTO THE NEXT

JAVA HAS ADDED EXACTLY THIS FUNCTIONALITY, USING SOMETHING CALLED "AGGREGATE OPERATIONS"

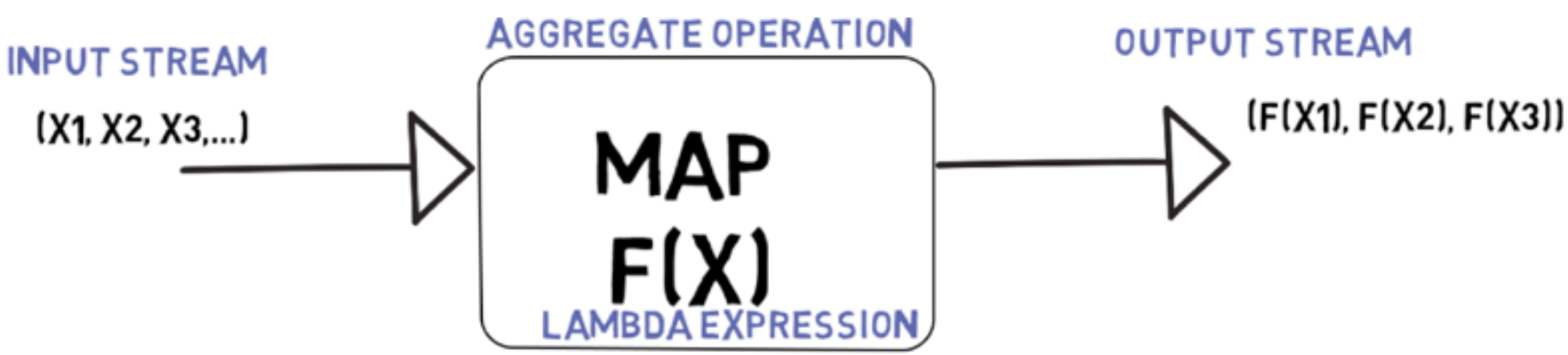
FILTER, MAP AND FOREACH ARE STANDARD AGGREGATE OPERATIONS IN FUNCTIONAL PROGRAMMING

JAVA PROVIDES A SIMPLE WAY TO USE THESE STANDARD FUNCTIONAL PROGRAMMING OPERATIONS TO COLLECTIONS LIKE LISTS, MAPS ETC

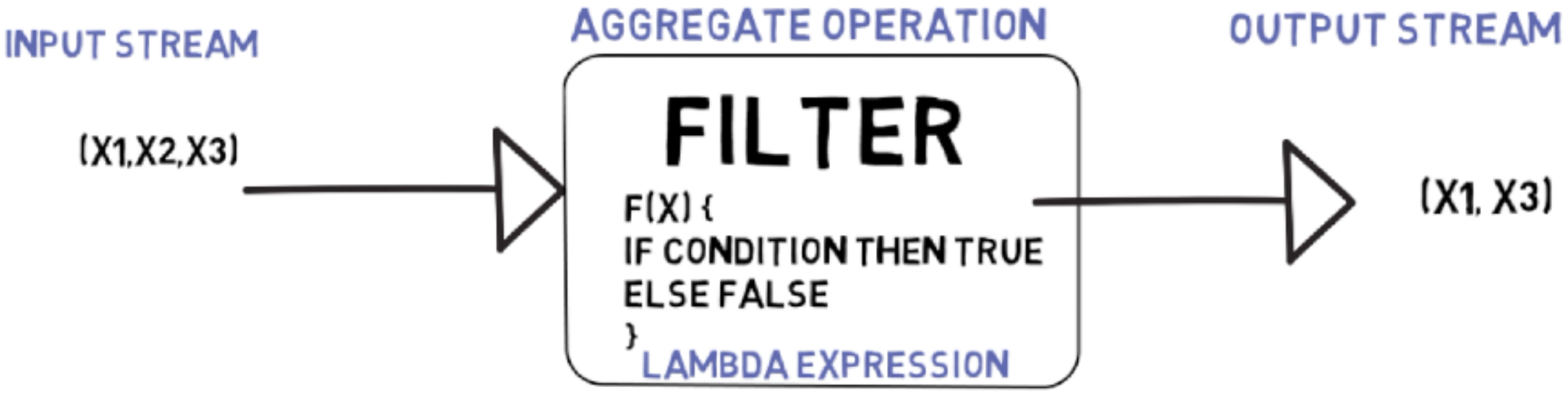
SIMPLY CALL THE ".STREAM()" METHOD
ON ANY COLLECTION TO GET AN OBJECT
OF TYPE STREAM, ON WHICH ON WHICH
AGGREGATE FUNCTIONS CAN BE APPLIED
IN SEQUENCE

THE JAVA "STREAM" OBJECT CAN BE IMAGINED
AS A STREAM OF VALUES, WHERE EACH VALUE
IS BEING SUBJECTED TO AN OPERATION
LIKE MAP, FOREACH, FILTER ETC

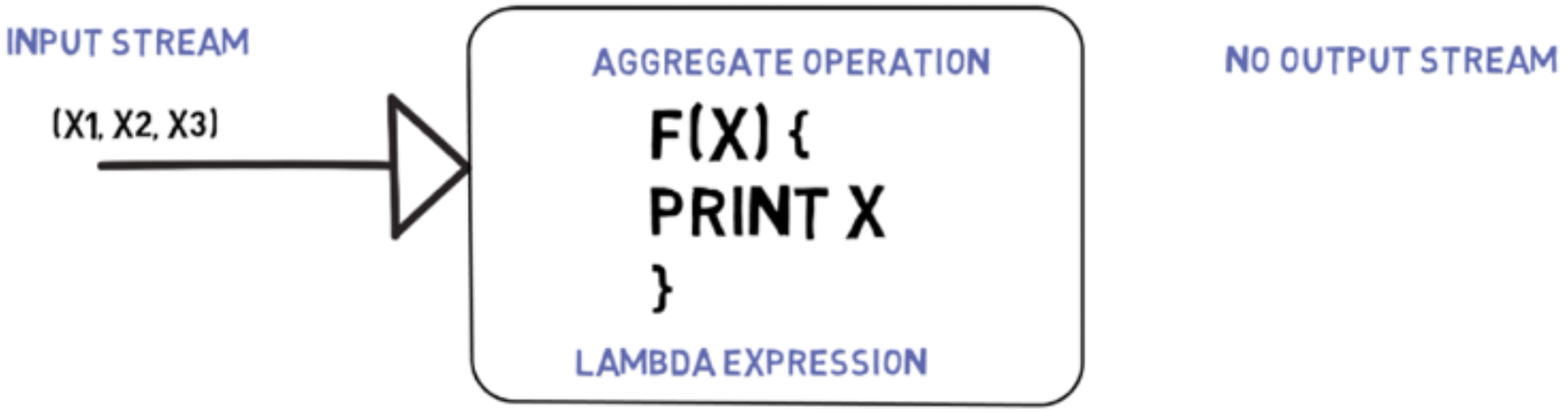
MAP IS AN AGGREGATE OPERATION THAT TAKES
A LAMBDA EXPRESSION, APPLIES IT TO
EVERY ELEMENT OF THE INPUT STREAM, AND
SENDS THE RESULTS OUT AS THE OUTPUT STREAM



FILTER IS AN AGGREGATE OPERATION THAT
TAKES IN A LAMBDA EXPRESSION THAT
ENCAPSULATES A CONDITION, APPLIES
IT TO EVERY ELEMENT OF AN INPUT STREAM;
EVERY ELEMENT THAT SATISFIES THE CONDITION
IS PLACED ON THE OUTPUT STREAM



FOREACH IS AN AGGREGATE OPERATION THAT TAKES A LAMBDA EXPRESSION AND APPLIES TO EACH ELEMENT OF AN INPUT STREAM, BUT DOES NOT PRODUCE AN OUTPUT STREAM



FOR-EACH IS USED FOR OPERATIONS LIKE PRINTING TO SCREEN OR SAVING TO FILE, WHERE IT MAKES NO SENSE TO PRODUCE AN OUTPUT STREAM