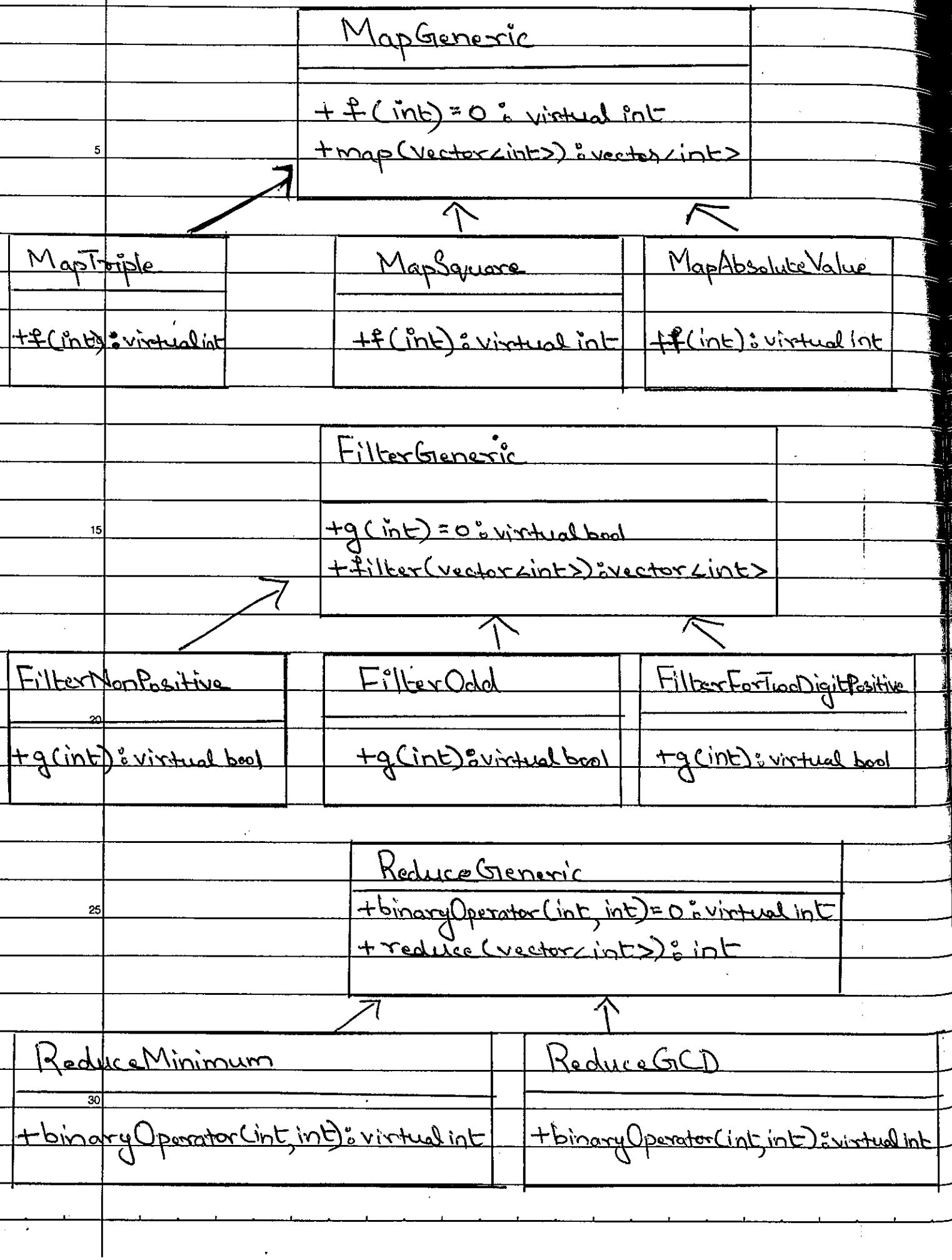
UML DIAGRAM



Description :-

Part 1 - Map

MapGeneric

* f(int) = 0 → A pure virtual function which will not include a definition for this function in the base class and the base class (MapGeneric) and can not be instantiated. It specifies the operation needed to map onto a list and is made private method of the class.
* map(vector<int>) → It takes a vector as the input and returns the resulting vector after mapping. The function should be called recursively to implement the map function

MapTriple

* f(int) → The function returns 3 times the number given as input.

MapSquare

* f(int) → The function returns the square of the numbers given as input.

MapAbsoluteValue

f(int) → The function returns the absolute value of a number given as input. Eg : Absolute value of -3 will be 3 and the absolute value of 3 will be 3.

Part 2 - Filter

FilterGeneric

* g(int) = 0 → This is a pure virtual method that is private and specifies the operation to be mapped onto a list and will be overridden later in the derived classes to deliver specific operations.
* filter(vector<int>) → It takes a vector as input and returns the resulting vector after filtering recursively.

FilterNonPositive

* q(int) → The function returns true if and only if the given value is not- positive.

FilterOdd

* q(int) → The function returns true if and only if the given value is a odd value.

FilterForTwoDigitPositive

* q(int) → The function returns true if and only if the given value is a 2 digit positive number.

Part-3- Reduce

ReduceGeneric

* binaryOperator(int, int) = 0 → This is a private method that specifies the operator. This method is overridden later in the derived classes to delives specific map operations and is a pure virtual function.
* reduce(vector<int>) → It takes a vector as the input and returns the result of reduce. Recursion is used to implement this reduce function.

ReduceMinimum

* binaryOperator(int, int) → This function returns the smaller value between between the given 2 integers values.

ReduceGCD

* binaryOperator(int, int) → This function returns the Greatest Common Divisor value between the given 2 integer values recursively.

P.T.O

Testing

1) Input

6, -15, 53, -16, 73, 128, 105, 104, -71, -179, 102, 12, 25, -145, -199, -156, -186, 43, -189

Output

45 45

2) Input

157, -24, -123, -81, 200, 157, 84, 67, -83, -60, -72, 192, -25, -20, -50, -181, -70, -23, -123

Output

69 3

3) Input

-21, 91, 46, 74, -44, 149, -192, 41, -9, -32, -133, 137, 178, -4, 119, -9, -111, -144, -184, -33

Output

27 9