Lambda expression

Def.: lambda expression is a concise way to represent an anonymous function. It helps to reduce boilerplate code when implementing functional interfaces

Currently we are writing detail code/step by step code is called verbose code.

* But in lambda expression we write code in short form but we achieve same operation.
* By using lambda expression we create anonymous functions.(it also method)
* In functional programing we can pass one function into another function.
* Lambda expression are used to implements only functional interface.

Example:

* In interface if we have many abstract method then while proceeding with lambda expression may get confuse to which abstract method should create anonymous function.
* For suppose in functional interface we have only one abstract method and if we write lambda expression know in functional interface have only one abstract method and for that method it create function.
* No need to implement the interface in class while we are proceeding with lambda expression.
* In side main method we can directly write.
* Syntax: (arguments), -> arrow, {method body};
* () //parentheses
* (Int a, int b) //a,b are called parameters.
* (5, 10) //5 and 10 are called arguments.
* If we have only one statement then no need of '{}'.
* We have only one parameter then no need of parentheses.

EG: speed -> {}

* If we have two parameters then we required parentheses.

E.g. :( int speed, int model) -> {}

* No need to specify data types also. Java can easily identify what type of data type it is.

E.g. :( speed, model) -> {}



