LAMBDA EXPRESSION

INTRODUCTION

- Lambda expressions are used for defining anonymous expressions or nameless methods or functions.
- Lambda expressions are defined with the help of interfaces.
- If a interface have single abstract method then it is called the functional interface.
- The lambda expression is '->'.
- These lambda expressions are very powerful and very useful and are very handy and easy for programmers.

PARAMETERS

- A method can take multiple parameters.
- Return keyword may not used in case of lambda expression.
- They contain just the expressions like methods.
- The lambda expressions may have either no parameters or one or multiple parameters.

CAPTURE IN LAMBDA EXPRESSION

- One can have multiple statements in lambda expressions.
- Variables can be declared inside the lambda expression itself.
- Lambda expressions can have local variable also.
- These expressions can access the local variables or capture local variables if they are final or they cannot be modified.

- Lambda expressions can even capture instance variables they may or may not be final.
- The lambda expressions are similar to inner classes.
- Lambda expression can be passed as a method as an object as it is used to define a method.
- When a method is taking a functional interface as parameter then you can pass lambda expression to that method.

METHOD REFERENCE

- Method references are either created or defined using functional interface.
- :: ‘scope resolution operator in c/c++’ this is the operator used for the referencing.
- In java ‘: :’ is used for method reference.
- Any method can be called or referred to by the functional interface with a single method.
- To non-static members method are assigned using objects.
- Constructor of any class can be assigned as method reference.
- The above method is used to write compact code.
- Method referencing is more like polymorphism.