Instructions for Practical Exercise

1. Use the Java_adv_classes_lambda folder of previous part. This folder will contains code files for each PE that you will do for the Java Inheritance and Common Classes PE.
2. Name code files corresponding to PE numbers. For example, Pe1.java is the code file for Practical Exercise 1 (PE 1)
3. Push your project to git

Practical Exercise: Java Advance Classes and Lambda Expressions – Part 4

PE 1: 1. Create a class with a non-default constructor (one with arguments) and no default constructor (no "no-arg" constructor). Create a second class that has a method that returns a reference to an object of the first class.Create the object that you return by making an anonymous inner class that inherits from the first class.

PE 2. Write a class named Outer that contains an inner class named Inner. Add a method to Outer that returns an object of type Inner. In main( ), create and initialize a reference to an Inner.

PE 3. Write a program to implement reverseSelector( ) which provides the following output 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 .

a. Create an interface named selector with methods end, current, next.

b. Create an outer class named Sequence which accepts the size from which you can reverse the numbers,

c. Create an inner class named ReverseSequenceSelector which implements the Selector

interface and try to reverse the number.

d. finally implement every classes in main and run the code.

Input: 15

Output : 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

PE 4. Create an interface named Hand with a method facepalm, and implement that interface by defining in an inner class named Palm within a method of the outer class, which returns a reference to your interface.

PE 5. Write a program to assign a lambda expression to its functional interface. Then we execute the lambda expression by calling the method defined in the functional interface and pass in a parameter.

Input: Java Lambda Expression Demo

Output : 27

PE 6. Write a program to create interface with single method and use it as lambda expression to multiply the given value by 2. When creating the lambda expression we declare the type of the parameter s1 to have the Integer type.

Input: 4

Output: 1- Result : 8

PE 7. Write a program to implement lambda expression which determines if one number is a factor of another along with the help of an interface with a method named test which accepts 2 integer.

Input : 10,2

Output :2 is a factor of 10

Input : 10,3

Output : 3 is not a factor of 10