Lambda expression assignments

1. Write an application to perform basic arithmetic operations like add, subtract, multiply & divide. You need to define a functional interface first.

```java
package LambdaAssignment;
public class Lambda1 {
        interface lambda{
            int operation(int x,int y);
            }
            public static double addition(int num1,int num2) {
            lambda get=(a,b) -> (a+b);
            return (double)(get.operation(num1,num2));
            }
            public static double subtraction(int num1,int num2) {
            lambda get=(a,b) -> (a-b);
            return (double)(get.operation(num1,num2));
            }
            public static double multiplication(int num1,int num2) {
            lambda get=(a,b) -> (a*b);
            return (double)(get.operation(num1,num2));
            }
            public static double division(int num1,int num2) {
            lambda get=(a,b) -> (a/b);
            return (double)(get.operation(num1,num2));
            }
            public static void main(String[] args) {
                int x=12, y=6;
                System.out.println(addition(x,y));
                System.out.println(subtraction(x,y));
                System.out.println(multiplication(x,y));
                System.out.println(division(x,y));
                }
    }
```

```
<terminated> Lambda1 [Java Application] C:\Users\valaanus\
18.0
6.0
72.0
2.0
```

3. Use the functional interfaces Supplier, Consumer, Predicate & Function to invoke built-in methods from Java API.

```java
package LambdaAssignment;
import java.util.function.Consumer;
import java.util.function.Function;
import java.util.function.Supplier;
public class Lambda3 {
         static void modifyValue(Consumer<Product> c, Product p1) {
          c.accept(p1);
          System.out.println("updated value:"+p1.getId());
      }
                static class Product{
                    private String name;
                    private int id;
                    public Product(String name, int id) {
                        super();
                        this.name=name;
                        this.id=id;
                    }
                    public String getName() {
                        return name;
                    }
                    public void setName(String name) {
                        this.name=name;
                    }
                    public int getId() {
                        return id;
                    }
                    public void setId(int id) {
                        this.id=id;
                    }
                }
                static <E> void display(Supplier<E> s) {
                    System.out.println(s.get());
                }
                static int returnIdByTen(int n, Function<Integer, Integer> fun) {
                    int res=fun.apply(n);
                    return res;
```

```
37                  }
38                  public static void main(String[] args) {
39                      Product p = new Product("abc", 154);
40                      display(()-> p.getId());
41                      display(()->p.getName());
42                      Consumer<Product> updateId=per->per.setId(123);
43                      updateId.accept(p);
44                      modifyValue(updateId, p);
45                      int n=p.getId();
46                      System.out.println(returnIdByTen(n, f->f+10));
47                  }
48  }
```

```
<terminated> Lambda3 [Java Application] C:\Users\valaanus\.p2\pool\plugins\org.
154
abc
updated value:123
133
```

4. Remove the words that have odd lengths from the list. HINT: Use one of the new methods from JDK 8. Use removeIf() method from Collection interface.

```
1  package LambdaAssignment;
2  import java.util.ArrayList;
3  public class Lambda4 {
4          public static ArrayList<String> removeOddLength(ArrayList<String> employeeList)
5          {
6              employeeList.removeIf(n->n.length()%2 != 0);
7              return employeeList;
8          }
9          public static void main(String[] args) {
10             ArrayList<String> arr = new ArrayList<>();
11             arr.add("abc");
12             arr.add("abcd");
13             arr.add("abcde");
14             arr.add("abcdef");
15             arr.add("abcdefg");
16             arr.add("abcdefgh");
17             ArrayList<String> arr1=removeOddLength(arr);
18             for(String str:arr1) {
19                 System.out.println(str);
20             }
21         }
22 }
```

```
<terminated> Lambda4 [Java Application] C:\Users\valaanus\.p2\pool\plugins\org
abcd
abcdef
abcdefgh
```

8. Create a new thread that prints the numbers from the list. Use class Thread & interface Consumer.

```java
package LambdaAssignment;
import java.util.ArrayList;
public class Lambda8 {
        public static void main(String[] args) {
            ArrayList<Integer> arr = new ArrayList<>();
            arr.add(1);
            arr.add(2);
            arr.add(3);
            arr.add(4);
            arr.add(5);
            Thread myThread = new Thread( () -> arr.forEach(n -> System.out.println(n)));
            myThread.start();
        }
}
```

```
<terminated> Lambda8 [Java Application] C:\Users\valaanus\.p2\pool\plugins\org.
1
2
3
4
5
```

7. Convert every key-value pair of the map into a string and append them all into a single string, in iteration order. HINT: Use Map.entrySet() method & a StringBuilder to construct the result String.

```java
package LambdaAssignment;
import java.util.Map.Entry;
import java.util.HashMap;
import java.util.Set;
public class Lambda7 {
        public static String convertKeyValueToString(HashMap<String, Integer> map) {
            String str="";
            Set<Entry<String, Integer>> s=map.entrySet();
            for(Entry<String, Integer> e : s) {
                str += e.getKey()+" ";
                str += e.getValue().toString()+" ";
            }
            return str;
        }
        public static void main(String[] args) {
            HashMap<String, Integer> map = new HashMap<>();
            map.put("abc", 1);
            map.put("def", 2);
            map.put("ghi", 3);
            System.out.println(convertKeyValueToString(map));
        }
}
```

```
<terminated> Lambda7 [Java Application] C:\Users\valaanus\.p2\pool\plugins\org
abc 1 def 2 ghi 3
```