**Lambda Assignment**

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

**Description:-**

Define Functional Interface and write a program to perform arithmetic operations like add, subtract, multiply and divide using functional interface.

**Examples:**

**Input:-**13       5

**Output:-**

18.0      //Addition of 13 and 5

                                       8.0        //Subtraction of 13 and 5

                                       65.0      // Multiplication of 13 and 5

                                       2.6        //Division of 13 and 5

**FunctionalInterface:-**

A functional interface is an interface that contains only one abstract method. They can have only one functionality to exhibit. From Java 8 onwards, lambda expressions can be used to represent the instance of a functional interface. A functional interface can have any number of default methods. Runnable, ActionListener, Comparable are some of the examples of functional interfaces.

**Specifications:**

public class Assignment4Q1 {  
    public double addition(int num1,int num2){}  
    public double subtraction(int num1,int num2){}  
    public double multiplication(int num1,int num2){}  
    public double division(int num1,int num2){}  
    public static void main(String[] args) {}  
}

**Code:**

package LambdaAssignment;

interface Arithmetic{

public double ArithmeticOperation (int num1,int num2);

}

public class Assignment4Q1 {

public double addition(int num1,int num2){

Arithmetic add = (int n1,int n2)-> n1+n2;

return add.ArithmeticOperation(num1,num2);

}

public double subtraction(int num1,int num2){

Arithmetic sub = (int n1,int n2)-> n1-n2;

return sub.ArithmeticOperation(num1,num2);

}

public double multiplication(int num1,int num2){

Arithmetic mul = (int n1,int n2)-> n1\*n2;

return mul.ArithmeticOperation(num1,num2);

}

public double division(int num1,int num2){

Arithmetic div= (int n1,int n2)-> (double) n1/n2;

return div.ArithmeticOperation(num1,num2);

}

public static void main(String[] args) {

Assignment4Q1 assignment4Q1 = new Assignment4Q1();

System.***out***.println(assignment4Q1.addition(10,20));

System.***out***.println(assignment4Q1.subtraction(20,10));

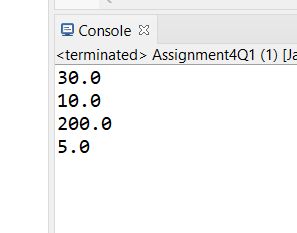
System.***out***.println(assignment4Q1.multiplication(10,20));

System.***out***.println(assignment4Q1.division(20,4));

}

}

**Output:**



[**Q2**](https://adapt.in.capgemini.com/mod/vpl/view.php?id=2156)**. Write an application using lambda expressions to print Orders having 2 criteria implemented: 1) order price more than 10000 2) order status is ACCEPTED or COMPLETED.**

**Description:**

Write a program in such a way that it has a method which returns the list of orders satisfying the 2 conditions mentioned in the question.

**Specifications:**

import java.util.ArrayList;  
  
public class Assignment4Q2 {  
  
    private int totalPrice;  
    private String status;  
  
    public static ArrayList<Assignment4Q2> listOfOrders(ArrayList<Assignment4Q2> orders) {}  
    public static void main(String[] args) {}  
}

**Code:**

package LambdaAssignment;

import java.util.ArrayList;

public class Assignment4Q2 {

public Assignment4Q2(int totalPrice, String status) {

this.totalPrice = totalPrice;

this.status = status;

}

public int getTotalPrice() {

return totalPrice;

}

public void setTotalPrice(int totalPrice) {

this.totalPrice = totalPrice;

}

public String getStatus() {

return status;

}

public void setStatus(String status) {

this.status = status;

}

private int totalPrice;

private String status;

public static ArrayList<Assignment4Q2> listOfOrders(ArrayList<Assignment4Q2> orders) {

ArrayList<Assignment4Q2> listOrders = new ArrayList<>();

for(Assignment4Q2 order:orders){

if((order.getStatus().equals("ACCEPTED") || order.getStatus().equals("COMPLETED"))&& order.getTotalPrice()>1000){

System.***out***.println("Inside if");

Assignment4Q2 or = new Assignment4Q2(order.getTotalPrice(),order.getStatus());

listOrders.add(or);

}

}

return listOrders;

}

public static void main(String[] args) {

Assignment4Q2 a1 = new Assignment4Q2(10001,"ACCEPTED");

Assignment4Q2 a2 = new Assignment4Q2(1000,"ACCEPTED");

Assignment4Q2 a3 = new Assignment4Q2(10500,"COMPLETED");

Assignment4Q2 a4 = new Assignment4Q2(100,"ACCEPTED");

Assignment4Q2 a5 = new Assignment4Q2(10002,"ACCEPTED");

Assignment4Q2 a6 = new Assignment4Q2(10005,"REJECTED");

Assignment4Q2 a7 = new Assignment4Q2(50000,"COMPLETED");

ArrayList<Assignment4Q2> list = new ArrayList<>();

list.add(a1);

list.add(a2);

list.add(a3);

list.add(a4);

list.add(a5);

list.add(a6);

list.add(a7);

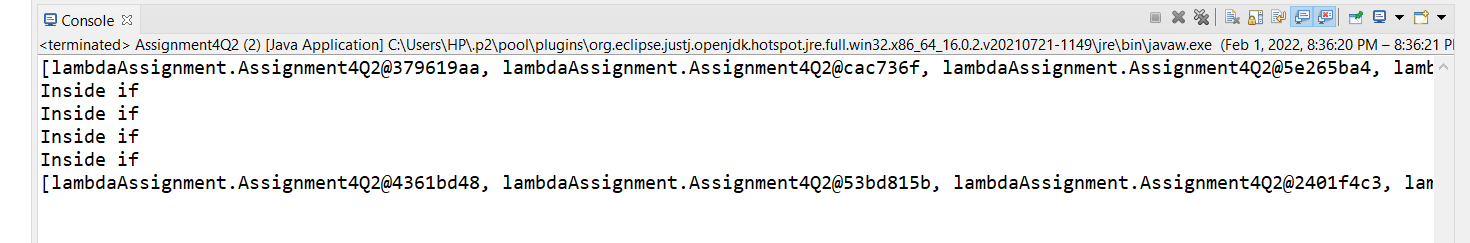
System.***out***.println(list);

System.***out***.println(*listOfOrders*(list));

}

}

**Output:**



[**Q3**](https://adapt.in.capgemini.com/mod/vpl/view.php?id=2261)**. Use the functional interfaces Supplier, Consumer, Predicate & Function to invoke built-in methods from Java API.**

**Description:**

Write a program using the Java API’s mentioned in the question.

**Specifications:**

public class Assignment4Q3 {  
    static void modifyValue(){  }  
    static class Product { }  
    static void display() { }  
    public static void main(String[] args) {  
}

**Code:**

package LambdaAssignment;

import java.util.function.Function;

public class Assignment4Q3 {

static int modifyValue(int n,Function fun) {

return (int)fun.apply(n);

}

static class Product {

}

static void display(int r) {

System.***out***.println(r);

}

public static void main(String[] args) {

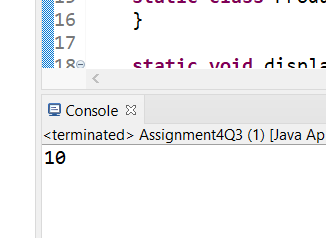
int r =*modifyValue*(5,val->(Integer)val+ 5);

*display*(r);

}

}

**Output:**



[**Q4**](https://adapt.in.capgemini.com/mod/vpl/view.php?id=2262)**. 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.**

**Description:-**

Write a program using java 8 features which can remove the odd length words from the list.

**Specifications:**

public class Assignment4Q4 {  
    public ArrayList<String> removeOddLength(ArrayList<String> employeeList){}  
    public static void main(String[] args) { }  
}

**Code:**

package LambdaAssignment;

import java.util.ArrayList;

import java.util.function.Function;

import java.util.function.Predicate;

import java.util.stream.Collectors;

interface RemoveDup

{

public int removeDup(String str);

}

public class Assignment4Q4 {

public ArrayList<String> removeOddLength(ArrayList<String> employeeList){

ArrayList<String> list = new ArrayList<>();

Predicate<String> filterEmp = (p)-> {

if(Math.*floorMod*(p.length(),2)==0)

return true;

else

return false;

};

employeeList.stream()

.filter(filterEmp)

.forEach(p->list.add(p));

return list;

}

public static void main(String[] args) {

ArrayList<String> employeeList = new ArrayList<>();

employeeList.add("TomJones");

employeeList.add("Harry");

employeeList.add("EthanHardy");

employeeList.add("NancySmith");

employeeList.add("Deborah");

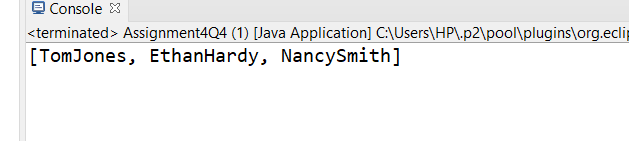
Assignment4Q4 assignment4Q4 = new Assignment4Q4();

System.***out***.println(assignment4Q4.removeOddLength(employeeList));

}

}

**Output:**



[**Q5**](https://adapt.in.capgemini.com/mod/vpl/view.php?id=2153)**. Create a string that consists of the first letter of each word in the list of Strings provided. HINT: Use Consumer interface & a String Builder to construct the result.**

**Description:**

Write a java program using StringBuilder and Consumer interface which will return a string. The returned string should consistes of the first let of each word in the list of words.

**Specifications:**

public class Assignment4Q5 {  
    List<String> list = Arrays.*asList*("alpha", "bravo", "charlie", "delta", "echo", "foxtrot");  
  
    public static void main(String[] args) { }  
  
    public static String processWords(List<String> list) {}  
}

**Code:**

package LambdaAssignment;

import java.util.Arrays;

import java.util.List;

import java.util.function.Consumer;

import java.util.stream.Stream;

public class Assignment4Q5 {

List<String> list = Arrays.*asList*("alpha", "bravo", "charlie", "delta", "echo", "foxtrot");

public static void main(String[] args) {

Assignment4Q5 assignment4Q5 = new Assignment4Q5();

*processWords*(assignment4Q5.list);

}

public static String processWords(List<String> list) {

StringBuilder words = new StringBuilder();

Consumer<String> consumer = p-> words.append(p.charAt(0));

list.stream().forEach(consumer);

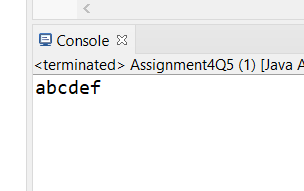
System.***out***.println(words);

return words.toString();

}

}

**Output:**



[**Q6**](https://adapt.in.capgemini.com/mod/vpl/view.php?id=2145)**. Replace every word in the list with its upper case equivalent. Use replaceAll() method & Unary Operator interface.**

Using replaceAll() method and Unary Operator interface write a java program which replaces evry word in the list with its upper case equivalent.

**Specifications:**

public class Assignment4Q6 {  
    public static void main(String[] args) {}  
    public List<String> convertToUpperCase(List<String> list) {}    
}

**Code:**

package LambdaAssignment;

import java.util.ArrayList;

import java.util.Arrays;

import java.util.List;

import java.util.function.Consumer;

public class Assignment4Q6 {

public static void main(String[] args) {

List<String> list = Arrays.*asList*("alpha", "bravo", "charlie", "delta", "echo", "foxtrot");

Assignment4Q6 assignment4Q6 = new Assignment4Q6();

System.***out***.println(assignment4Q6.convertToUpperCase(list));

}

public List<String> convertToUpperCase(List<String> list) {

List<String> ToUpper = new ArrayList<>();

Consumer<String> consumer = (str)->ToUpper.add(str.toUpperCase());

list.stream()

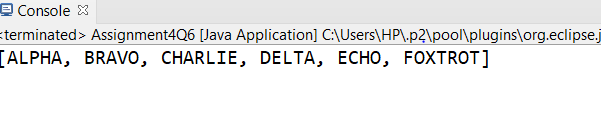
.forEach(consumer);

return ToUpper;

}

}

**Output:**



[**Q7**](https://adapt.in.capgemini.com/mod/vpl/view.php?id=2146)**. 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.**

**Description:-**

Write a java program using Map.entrySet() method & a StringBuilder which will return a string by appending all the key value pairs of a map into a single string ,in insertion order.

**Specifications:**

public class Assignment4Q7 {  
    public static void main(String[] args) {}  
    public String convertKeyValueToString(HashMap<String, Integer> map) {}  
}

**Code:**

package LambdaAssignment;

import java.util.HashMap;

import java.util.Iterator;

import java.util.Map;

import java.util.Set;

import java.util.function.Consumer;

public class Assignment4Q7 {

public static void main(String[] args) {

HashMap<String,Integer> map = new HashMap<>();

map.put("a",4);

map.put("b",2);

map.put("c",3);

Assignment4Q7 assignment4Q7 = new Assignment4Q7();

assignment4Q7.convertKeyValueToString(map);

}

public String convertKeyValueToString(HashMap<String, Integer> map) {

StringBuilder str = new StringBuilder();

Consumer<Map.Entry> consumer=(p)->str.append(p.getKey()).append(p.getValue());

Set set = map.entrySet();

set.stream().forEach(consumer);

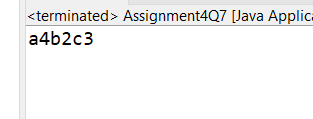
System.***out***.println(str);

return str.toString();

}

}

**Output:**



[**Q8**](https://adapt.in.capgemini.com/mod/vpl/view.php?id=2147)**. Create a new thread that prints the numbers from the list. Use class Thread & interface Consumer.**

**Description:-**

Write a java program which will print the list of number using Thread and interface Consumer.

**Specifications:**

public class Assignment4Q8 {}

**Code:**

package LambdaAssignment;

import java.util.ArrayList;

import java.util.Arrays;

import java.util.List;

import java.util.function.Consumer;

public class Assignment4Q8 {

public static void main(String[] args) {

List<Integer> list1 = Arrays.*asList*(1,2,3,4,5,6,7,8);

Consumer<List<Integer> > dispList = list -> list.stream().forEach(a -> System.***out***.print(a + " "));

Thread t = new Thread(()->dispList.accept(list1));

t.start();

}

}

**Output:**

