1. Write a application to develop basic arithmetic operation.

package lam;

public class exp {

public static void main(String[] args) {

arthmetic mylambda= (int a,int b) ->a+b;

System.out.println("Addition:"+(mylambda.foo(10,10)) );

arthmetic mylambda1= (int a,int b) ->a-b;

System.out.println("Subtraction:"+(mylambda1.foo(20, 10)));

arthmetic mylambda2= (int a,int b) ->a\*b;

System.out.println("Multiplication:"+(mylambda2.foo(20, 10)));

arthmetic mylambda3= (int a,int b) ->a/b;

System.out.println("Division:"+(mylambda3.foo(20, 10)));

}

interface arthmetic{

int foo(int a, int b);

}

}

Output:

Addition:20

Subtraction:10

Multiplication:200

Division:2

1. Write an application using lambda expressions to print the order.

package lam;

import java.util.Scanner;

public class exp1 {

public static void main(String[] args)

{

order mylambda = (int a) -> {

if(a >10000)

{

System.out.println("Accepted");

}

else

{

System.out.println("Not Accepted");

}

return a;

};

System.out.println("order amount:" +(mylambda.foo(100000)));

}

interface order

{

int foo(int a);

}

}

Output:

Accepted

order amount:100000

1. Remove the words that have odd lengths from the list.

package lam;

import java.util.ArrayList;

public class exp4 {

public static void main(String[] args)

{

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

words.add("hi");

words.add("Hey");

words.add("Hello");

words.add("good");

words.add("bad");

words.add("three");

words.removeIf(n->(n.length()%2!=0));

for(String i:words)

{

System.out.println(i);

}

}

}

Output:

hi

good

1. Create a string that consists of the first letter of each word in the list provided.

**package** lam;

**import** java.util.ArrayList;

**import** java.util.function.Consumer;

**public** **class** exp5 {

**public** **static** **void** main(String[] args)

{

ArrayList<String>words=**new** ArrayList<String>();

words.add("hi");

words.add("Hey");

words.add("Hello");

words.add("good");

words.add("bad");

words.add("three");

Consumer <String> print=(str)->System.***out***.println("the first letter of strings:"+str.charAt(0));

words.forEach(print);

}

}

Output:

the first letter of strings:h

the first letter of strings:H

the first letter of strings:H

the first letter of strings:g

the first letter of strings:b

the first letter of strings:t

1. Use the functional interfaces supplier,consumer and predicate

import java.util.ArrayList;

import java.util.function.Consumer;

import java.util.function.Predicate;

import java.util.function.Supplier;

public class exp3 {

public static void main(String[] args)

{

//predicate

Predicate<Integer>gt=a->(a>10);

System.out.println("predicate:" + gt.test(20));

//supplier

String str="HI";

Supplier<Integer> supplier=()->str.length();

System.out.println("Supplier:" +supplier.get());

//Consumer

Consumer<String>print=a->System.out.println("Consumer:"+a);

print.accept("HI");

}

}

Output:

predicate:true

Supplier:2

Consumer:HI

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

package lam;

import java.util.ArrayList;

import java.util.function.Consumer;

//import java.util.function.Consumer;

import java.util.function.UnaryOperator;

public class exp5 {

public static void main(String[] args)

{

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

words.add("hi");

words.add("Hey");

words.add("Hello");

words.add("good");

words.add("bad");

words.add("three");

Thread obj=new Thread(()->System.out.println(words));

obj.run();

}

}

Output:

[hi, Hey, Hello, good, bad, three]

1. 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 String Builder to construct the result String.

package lam;

import java.util.HashMap;

import java.util.Map;

import java.util.stream.Collectors;

public class exp7 {

public static void main(String[] args)

{

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

map.put("Hi", "Hello");

map.put("Good", "bad");

map.put("sweet","salt");

map.put("sing","sour");

String s = map.entrySet().stream().map((entry) ->" " + entry.getKey() + " " + entry.getValue().replaceAll("\s ", "\\s ") + " ").collect(Collectors.joining(" "));

System.out.println(s);

}

Output:

Hi Hello sing sour Good bad sweet salt

1. Replace every word in the list with its upper case equivalent. Use replaceAll method & Unary Operator interface.

package lam;

import java.util.ArrayList;

public class exp6 {

public static void main(String[] args)

{

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

words.add("hi");

words.add("Hey");

words.add("Hello");

words.add("good");

words.add("bad");

words.add("three");

words.replaceAll(s->s.toUpperCase());

for(String i:words)

{

System.out.println("print:" +i);

}

}

}

Output:

print:HI

print:HEY

print:HELLO

print:GOOD

print:BAD

print:THREE