Core Java Assignment 8

# Assignments on Lambda Expression

1. Write a application to develop basic arithmetic operation.

**package** Lambda;

**interface** Arithmetic {

**int** operation(**int** a, **int** b);

}

**public** **class** LambdaExample {

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

Arithmetic addition = (**int** a, **int** b) -> (a + b);

System.***out***.println("Addition = " + addition.operation(5, 6));

Arithmetic subtraction = (**int** a, **int** b) -> (a - b);

System.***out***.println("Subtraction = " + subtraction.operation(5, 3));

Arithmetic multiplication = (**int** a, **int** b) -> (a \* b);

System.***out***.println("Multiplication = " + multiplication.operation(4, 6));

Arithmetic division = (**int** a, **int** b) -> (a / b);

System.***out***.println("Division = " + division.operation(12, 6));

}

}

**Output:**

Addition = 11

Subtraction = 2

Multiplication = 24

Division = 2

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

**package** Lambda;

**import** java.util.Scanner;

**public** **class** Print {

**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(100001)));

}

**interface** Order

{

**int** foo(**int** a);

}

}

**Output:**

Accepted

Order amount:100001

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

**package** Lambda;

**import** java.util.ArrayList;

**public** **class** OddLength {

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

{

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

words.add("Good Morning");

words.add("Hello");

words.add("Hey");

words.add("Shubham");

words.add("Cute");

words.add("three");

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

**for**(String i:words)

{

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

}

}

}

**Output:**

Good Morning

Cute

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

**package** Lambda;

**import** java.util.ArrayList;

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

**public** **class** FirstLetter {

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

{

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

words.add("Good Morning");

words.add("Hello");

words.add("Hey");

words.add("Shubham");

words.add("Cute");

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:G

The first letter of strings:H

The first letter of strings:H

The first letter of strings:S

The first letter of strings:C

The first letter of strings:t

1. Use the functional interfaces supplier,consumer and predicate

**package** Lambda;

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

**import** java.util.function.Predicate;

**import** java.util.function.Supplier;

**public** **class** Interface {

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

{

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

System.***out***.println("Predicate:" + gt.test(50));

String str="Heyoo";

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

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

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

print.accept("Hello");

}

}

**Output:**

Predicate:true

Supplier:5

Consumer:Hello

1. Replace every word in the list with its upper case equivalent. Use replaceAll() method and UnaryOperator interface.

Ans- Here I created 2 class first is Replace.java and another is Main.java

Class1 : Replace.java

**package** Lambda;

**import** java.util.function.UnaryOperator;

**class** Replace **implements** UnaryOperator<String> {

**public** String apply(String str) {

**return** str.toUpperCase();

}

}

Class2 : Main.java

**package** Lambda;

**import** java.util.ArrayList;

**public** **class** Main {

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

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

list.add("one");

list.add("two");

list.add("three");

list.add("four");

list.add("five");

System.***out***.println("Contents of the list: " + list);

list.replaceAll(**new** Replace());

System.***out***.println("Contents of the list after replace operation:" + list);

}

}

**Output:**

Contents of the list: [one, two, three, four, five]

Contents of the list after replace operation:[ONE, TWO, THREE, FOUR, FIVE]

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 StringBulider to construct the result String.

**package** Lambda;

**import** java.util.HashMap;

**import** java.util.Map;

**import** java.util.stream.Collectors;

**public** **class** Append {

**public** **static** **void** main(String[] cmd\_lineParams) {

Map<String, String> map = **new** HashMap<>(5);

map.put("KTM", "1");

map.put("BMW", "2");

map.put("Kawasaki", "3");

map.put("Ducati", "6");

map.put("Indian", "5");

String s = map.entrySet().stream().map((entry) ->

"" + entry.getKey() + " \"" + entry.getValue().replaceAll("\"", "\\\\\"") + "\"")

.collect(Collectors.*joining*(" "));

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

}

}

**Output:**

Indian "5" Ducati "6" KTM "1" BMW "2" Kawasaki "3"

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

**package** Lambda;

**import** java.util.ArrayList;

**import** java.util.List;

**public** **class** Consumer {

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

List<Integer> l = **new** ArrayList<Integer>(){{

add(11);

add(21);

add(12);

add(3);

add(50);

}};

Thread mylambda = **new** Thread(()->System.***out***.println(l));

mylambda.run();

}

}

**Output:**

[11, 21, 12, 3, 50]