

Lambda

1

```
package lambda;

public class lambda1 {

    public static void main(String[] args) {

        Arithmetic add= (x,y)-> x+y;
        System.out.println("addition is: " +add.operation(10,5));
        Arithmetic sub= (x,y)-> x-y;
        System.out.println("subtraction is: " +sub.operation(10,5));
        Arithmetic mul= (x,y)-> x*y;
        System.out.println("multiplication is: " +mul.operation(10,5));
        Arithmetic div= (x,y)-> x/y;
        System.out.println("division is: " +div.operation(10,5));

    }

}

@FunctionalInterface
interface Arithmetic
{
    int operation(int x, int y);
}
```

```
addition is: 15
subtraction is: 5
multiplication is: 50
division is: 2
```

2

```
package lambda;

public class lambdaorder {

    int no;
    String status;
    int price;

    public lambdaorder(int no, String status, int price) {
        super();
        this.no = no;
        this.status = status;
        this.price = price;
    }

    public int getNo() {
        return no;
    }

    public void setNo(int no) {
        this.no = no;
    }

    public String getStatus() {
        return status;
    }

    public void setStatus(String status) {
        this.status = status;
    }

    public int getPrice() {
        return price;
    }

    public void setPrice(int price) {
        this.price = price;
    }

}
&
```

```
package lambda;

import java.util.List;
import java.util.ArrayList;
import java.util.stream.Stream;

public class order {
```

```

    public static void main(String[] args) {
        List<lambdaorder> list= new ArrayList<>();
        list.add(new lambdaorder(1, "completed", 5000));
        list.add(new lambdaorder(2, "accepted", 11000));
        list.add(new lambdaorder(3, "accepted", 9800));
        list.add(new lambdaorder(4, "completed", 15000));
        list.add(new lambdaorder(5, "accepted", 50000));

        System.out.println("status for orders having price greater than
10000");
        Stream<lambdaorder> store= list.stream().filter(p-> p.price >10000);
        store.forEach(lambdaorder -> System.out.println(lambdaorder.no+"
"+lambdaorder.price+" "+lambdaorder.status));
    }
}

```

```

status for orders having price greater than 10000
2 11000 accepted
4 15000 completed
5 50000 accepted

```

3

```

package lambda;

import java.util.Arrays;
import java.util.List;

public class predicate {

    public static void main(String[] args) {
        List<Integer> list= Arrays.asList(1,11,111,2,4,5,76,9);
        list.stream().filter(t-> t%2==0).forEach(t->
System.out.println("printing even: " +t));
    }
}

```

```

printing even: 2
printing even: 4
printing even: 76

```

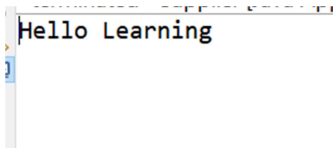
and

```
package lambda;

import java.util.function.Supplier;

public class supplier {

    public static void main(String[] args) {
        Supplier<String> store=() -> "Hello Learning";
        System.out.println(store.get());
    }
}
```



Hello Learning

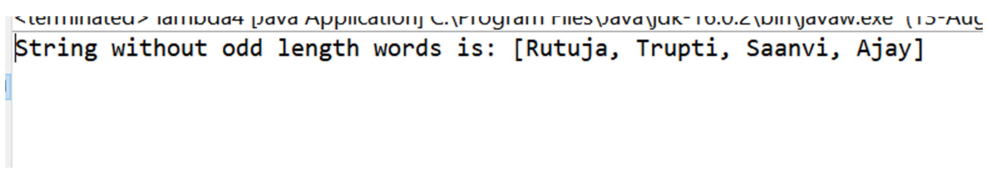
4

```
package lambda;

import java.util.List;
import java.util.ArrayList;
import java.util.Arrays;

public class lambda4 {

    public static void main(String[] args) {
        List<String> details= new ArrayList<>(Arrays.asList("Rutuja",
"Trupti", "Saanvi", "Kirti", "Priyank", "Ajay"));
        details.removeIf(i-> i.length()%2 !=0);
        System.out.println("String without odd length words is: " +details);
    }
}
```



<terminated> lambda4.java Application] C:\Program Files\Java\jdk-10.0.2\bin\javaw.exe (15-Aug
String without odd length words is: [Rutuja, Trupti, Saanvi, Ajay]

5

```
package lambda;

import java.util.ArrayList;
import java.util.Arrays;
import java.util.List;
import java.util.function.Consumer;

public class lambda5 {

    public static void main(String[] args) {
        List<String> details= Arrays.asList("Rutuja", "Trupti", "Saanvi",
        "Kirti", "Priyank", "Ajay");
        StringBuilder str= new StringBuilder();
        forEach(details, a-> str.append(a.charAt(0)));
        System.out.println(str);
    }
    static <String> void forEach(List<String> details, Consumer<String>
consumer)
    {
        for (String t: details)
        {
            consumer.accept(t);
        }
    }
}
```

RTSKPA

6

```
package lambda;

import java.util.Arrays;
import java.util.ArrayList;

public class lambda6 {

    public static void main(String[] args) {
        ArrayList<String> name= new ArrayList<>(Arrays.asList("Rutuja",
        "Trupti", "Saanvi", "Kirti", "Priyank", "Ajay"));
        System.out.println(name);
        name.replaceAll(a->a.toUpperCase());
        System.out.println(name);
    }
}
```

```
[Rutuja, Trupti, Saanvi, Kirti, Priyank, Ajay]  
[RUTUJA, TRUPTI, SAANVI, KIRTI, PRIYANK, AJAY]
```

7

```
package lambda;  
  
import java.util.Map;  
import java.util.HashMap;  
  
public class lambda7 {  
  
    public static void main(String[] args) {  
        Map<Integer, String> map= new HashMap <Integer, String>();  
        map.put(1, "pen");  
        map.put(2, "paper");  
        map.put(3, "scissors");  
  
        StringBuilder sb=new StringBuilder("");  
        for (Map.Entry m:map.entrySet())  
        {  
            sb.append("key " +m.getKey()+ " ");  
            sb.append("value " +m.getValue()+ " ");  
        }  
        System.out.println(sb);  
    }  
}
```

```
key 1 value pen key 2 value paper key 3 value scissors
```

8

```
package lambda;

import java.util.List;
import java.util.ArrayList;
import java.util.function.Consumer;

public class lambda8 extends Thread{
    public void run()
    {
        System.out.println("running...");
    }

    public static void main(String[] args) {
        lambda8 p=new lambda8();
        p.start();
        List<Integer> nos=new ArrayList();
        nos.add(1);
        nos.add(2);
        nos.add(1);
        nos.add(12);
        nos.add(11);

        Consumer<List<Integer>> print=list ->list.stream().forEach(a->
System.out.print(a+ " "));
        print.accept(nos);
    }
}
```

```
running...
1 2 1 12 11
```