

LAB-19

ABHINANDAN SINGH PARMAR

22BTRAD001

```
1 object CommonElementsSetExample {
2
3 def main(args: Array[String]): Unit = {
4
5 val set1 = Set(1, 2, 3, 4, 5)
6 val set2 = Set(6, 7, 8)
7
8 println("Set1: " + set1)
9 println("Set2: " + set2)
10 val common = set1.intersect(set2)
11 if (common.isEmpty){
12     println("There are no common elements")
13 }
14 else{
15     println("Common elements: " + common)
16 }
17 }
18 }
```

STDIN

Input for the program (Optional)

Output:

Set1: HashSet(5, 1, 2, 3, 4)
Set2: Set(6, 7, 8)
There are no common elements

```
1 object CommonElementsSetExample {
2 def main(args: Array[String]): Unit = {
3
4 val set1 = Set(1, 2, 3, 4, 5)
5 val set2 = Set(4, 5, 6, 7, 8)
6
7 println("Set1: " + set1)
8 println("Set2: " + set2)
9
10 val commonElements = set1.intersect(set2)
11
12 println("Common elements: " + commonElements)
13 }
14 }
```

STDIN

Input for the program (Optional)

Output:

Set1: HashSet(5, 1, 2, 3, 4)
Set2: HashSet(5, 6, 7, 8, 4)
Common elements: HashSet(5, 4)

CODE:

```
object CommonElementsSetExample {

  def main(args: Array[String]): Unit = {

    val set1 = Set(1, 2, 3, 4, 5)
    val set2 = Set(6, 7, 8)

    println("Set1: " + set1)
    println("Set2: " + set2)
    val common = set1.intersect(set2)
    if (common.isEmpty){
      println("There are no common elements")
    }
    else{
      println("Common elements: " + common)
    }
  }
}
```

Output:

```
Set1: HashSet(5, 1, 2, 3, 4)
Set2: Set(6, 7, 8)
There are no common elements
```

Github:

<https://github.com/AbhinandanSinghParmar/scala>