AASHUTOSH MISHRA
1NT22CS002

Array:

```
scala> var colors = new Array[String](7)
colors: Array[String] = Array(null, null, null, null, null, null, null)

scala> colors(0) = "RED"

scala> colors
res1: Array[String] = Array(RED, null, null, null, null, null, null)

scala> colors(1)="BLUE"
```

List:

```
scala> var Department = "CSE"::"ECE"::"ISE"::Nil
Department: List[String] = List(CSE, ECE, ISE)
```

map and foreach:

```
scala> val rows  = List(1, 2, 3, 4, 5)
rows: List[Int] = List(1, 2, 3, 4, 5)

scala> val numbers = List(1, 2, 3, 4, 5)
numbers: List[Int] = List(1, 2, 3, 4, 5)

scala> numbers.foreach(number => println(s"Number: $number"))
Number: 1
Number: 2
Number: 3
Number: 4
Number: 5

scala> numbers.foreach(println)
1
2
3
4
5

scala> val doubledNumbers = numbers.map(number => number * 2)
doubledNumbers: List[Int] = List(2, 4, 6, 8, 10)

scala> println(doubledNumbers)
List(2, 4, 6, 8, 10)
```

Lowercase:

```
scala> val originalString = "HeLlO wOrLd"
originalString: String = HeLlO wOrLd

scala> val lowercaseString = originalString.toLowerCase()
lowercaseString: String = hello world

scala> println(lowercaseString)
hello world

scala> val stringList = List("ApPlE", "BaNaNa", "OrAnGe")
stringList: List[String] = List(ApPlE, BaNaNa, OrAnGe)

scala> val lowercaseList = stringList.map(_.toLowerCase())
lowercaseList: List[String] = List(apple, banana, orange)

scala> println(lowercaseList)
List(apple, banana, orange)
```

Reduce Method:

```
scala> val numbers = List(1, 2, 3, 4, 5)
numbers: List[Int] = List(1, 2, 3, 4, 5)

scala> val sum = numbers.reduce((x, y) => x + y)
sum: Int = 15

scala> println(sum)
15

scala> numbers.reduce((x,y)=>x max y)
res1: Int = 5
```

Reduce Left and Right:

```
scala> val mul = List(2.0,1.4,6.7)
mul: List[Double] = List(2.0, 1.4, 6.7)

scala> val divide=(n : Double, m : Double)=>{
     | val result = n/m
     | result
     | }
divide: (Double, Double) => Double = <function2>


scala> mul.reduceLeft(divide)
res6: Double = 0.21321961620469082

scala> mul.reduceRight(divide)
```

```
res7: Double = 9.571428571428571
```

Iterator:

```
scala> val numbers = List(1, 2, 3, 4, 5).iterator
numbers: Iterator[Int] = non-empty iterator

scala> while (numbers.hasNext) {
     |   // Get the next element
     |   val number = numbers.next()
     |   println(number)
     | }
1
2
3
4
5


scala> val rangeIterator = (1 to 5).iterator
rangeIterator: Iterator[Int] = non-empty iterator

scala>

scala> while (rangeIterator.hasNext) {
     |   println(rangeIterator.next())
     | }
1
2
3
4
5
```

Pattern mathcing:

```
scala> val x = 3
x: Int = 3

scala> val result = x match {
     |   case 1 => "one"
     |   case 2 => "two"
     |   case 3 => "three"
     |   case _ => "other"
     | }
result: String = three
```