Agenda

```
Arrays
    Multidimensional
    Ragged
Variable Arity/Argument Method
String
StringBuffer & StringBuilder
String Tokenizer
enum
```

Single Dimensional Array (Demo01 & Demo02)

MultiDimensional Array (Demo03 & Demo04)

```
- For a multildimensional array we have array inside another array.
- Let us take an example of 2D Array in java
```

```java
        // primitive type
        int arr[][] = new int[5][3];

        //reference type
        Employee [][] empArr = new Employee[3][2];
```

Ragged Array (Demo05 & Demo06)

```
- It is an multidimensional array where column size is not fixed
```

```java
        // primitive type
        int arr[][] = new int[2][];
        arr[0] = new int[2];
        arr[1] = new int[4];

        //reference type
        Employee [][] empArr = new Employee[2][];
        empArr[0] = new Employee[2];
        empArr[1] = new Employee[4];
```

Variable Arity/Argument Method (Demo07)

```java
public void add(int ... num1) {
        int result = 0;
        for(int i:num1)
            result = result+i;
        System.out.println("Addition = "+result);
    }
```

String (Demo08 to Demo11)

```
String is a reference type in java
We can create a string value in two ways
```

```java
    String name1 = "sunbeam";//literal
    String name2 = new String("sunebam")// object
```

```
- If we create one of more reference with same literal as of previous then
both the references will point at same literal in the literal pool.
- If we cretae one more object with same value as that of previous object
then both the references will point at different objects in the memory
- We can check the literals for equality using == operator while for
objects we need to use equals() of string class
- If we use + operator to concat string then literal gets created
- If we use concat() to concat string then string object is created
- If we create a new String reference and use already created reference to
assign to it then a new String object is created.
- String is immutable. once created you cannot change its value
```

StringBuffer & StringBuilder (Demo12)

```
- StringBufffer and StringBuilder are both mutabale classes
- StringBuffer is threadsafe while StringBuilder is not.
-equals and hashcode methods are not overriden inside StringBuilder and
StringBuffer class
```

String Tokenizer (Demo13)

```
- If you want to split the string into token use this class.
- You have to cretae the object of this StringTokenizer class to call the
methods inside it.
```

enum (Demo14 to Demo16)

Labflow

```
Single Dimension Array
MultiDimension
***String
String Tokeninzer
enum
```