**Strings and Arrays in Java**  
   
i) String Handling in Java  
  
ii) Arrays in Java  
-------------------------------  
**i) String Handling in Java**  
  
a) What is String?   
> String is a sequence of characters written double quotes.  
------------  
Numbers  
-------------  
Integers - byte, short, int, long data types  
  
Floating point values/decimal values- float, double data types  
-----------------------------  
Character -char  
------------------  
Logical values  
  
boolean   
----------------------  
String -Object  
----------------  
  
> String may have Alphabets, Numbers and Special characters.  
  
Example:  
  
System.out.println("Selenium");//Selenium  
System.out.println("123Selenium");//123Selenium  
System.out.println("Selenium\*&123");//Selenium\*&123  
System.out.println("1234");//1234  
--------------------------------  
b) Create Strings  
   
String myTool ="Selenium";//String Variable  
String [] myTools ={"UFT", "Selenium", "LoadRunner", "RFT"}; //Array of Strings  
          
System.out.println(myTool);//Selenium  
          
for (String tool: myTools){ // Enhanced for loop  
System.out.println(tool);  
}  
------------------------------------  
c) Operations on Strings  
  
1) Concatenating Strings  
   
String str1 = "Selenium ";  
String str2 ="Testing";  
System.out.println(str1 + str2);//Selenium Testing  
System.out.println("Selenium" + (1 + 1));//Selenium2  
System.out.println("Selenium" + 1 + 1);//Selenium11  
System.out.println(1 + 1 + "Selenium");//2Selenium  
System.out.println("1" + 1 + "Selenium");//11Selenium  
System.out.println("Selenium" + " ");  
System.out.println(" " + "Selenium");  
  
String + String = Concatenation  
  
String + Integer = Concatenation  
  
Integer + Integer = Addition  
------------------------------------  
2) String Comparison  
   
In computer programming we have 2 types of comparison  
  
i) 2-way Comparison (true/false)  
  
ii) 3-way Comparison (0, > 0, < 0)  
--------------------------------  
a) String comparison using (==) Relational Operator  
  
It supports 2-way Comparison(true/false)  
--------------------------------  
b) Sting comparison using equals() method  
  
It supports 2-way Comparison(true/false)  
-----------------------------------  
c) Sting comparison using compareTo() method  
  
It supports 3-way Comparison (0, >0, <0)  
------------------------------------  
// A to Z (65 to 90) A= 65 , B=66….  
// a to z (97 to 122) a=97, b=98…..  
// 0 to 9 (48 to 57) 0=48 , 1=49  
  
Example:  
String str1 = "selenium";259  
String str2 = "SELENIUM";458-259-456String str3 = "SELENIUM";  
String str4 = "zselenium";  
          
//String Comparison using == Operator  
System.out.println(str1 == str2);//false  
System.out.println(str2 == str3);//true  
          
//String Comparison using equals() method  
System.out.println(str1.equals(str2));//false  
System.out.println(str2.equals(str3));//true  
          
//String Comparison using compareTo() method  
System.out.println(str1.compareTo(str2));//Greater than 0  
System.out.println(str2.compareTo(str3));//0  
System.out.println(str1.compareTo(str4));//Less than 0  
--------------------------------  
Result Criteria for 3-way comparison  
  
if str1 = str2 then 0  
  
if str1 > str2 then > 0  
  
if str1 < str2 then < 0  
----------------------------------------------  
**ii) Arrays in Java**  
  
a) What is Java Array?  
   
> Java Array is an Object that holds a fixed number of values of a single data type.  
  
> The length of Array is established when the Array is created.  
  
> Array length is fixed, index starts from zero to n-1.  
  
b) Creating of Arrays  
   
1st Method  
  
dataType arrayName []; //Creating Array  
  
arrayName = new dataType[size]; //Define Size   
  
arrayName[0]=value;//Assign value  
arrayName[1]=value;  
arrayName[2]=value;  
.  
.  
----------------------------------  
Example:  
   
int a [];  
a = new int[3];  
          
a[0]=10;  
a[1]=20;  
a[2]=30;  
System.out.println(a[0]);//10  
System.out.println(a[1] + a[2]);//50  
-------------------------------------  
//Assign values to elements that more than the length of Array (Run-Time Error)  
   
int a [];  
a = new int[3];  
          
a[0]=10;  
a[1]=20;  
a[2]=30;  
a[3]=40;//Out of Range(Run-Time Error)  
System.out.println(a[0]);//10  
System.out.println(a[1] + a[2]);//50  
------------------------------------  
//Assign values to some elements only (No Error)  
  
int a [];  
a = new int[3];  
          
a[1]=20;  
a[2]=30;  
System.out.println(a[1] + a[2]);//50  
------------------------------------------------------  
//If we assign invalid values (data type) -Syntax Error   
  
int a [];  
a = new int[3];  
a[0] =1.23; //Syntax Error   
a[1]=20;  
a[2]=30;  
System.out.println(a[1] + a[2]);//50  
------------------------------------------------          
2nd Method  
  
dataType [] arrayName= new dataType[length]; //Declare Array with length  
  
arrayName[index] = value; //Assign value  
  
Example:  
  
int [] abc = new int [4];  
abc[0] =10;  
System.out.println(abc[0]);    //10      
------------------------------  
**3rd Method (Declare Array and Assign values)**  
   
dataType [] arrayName = {value1, value2, value3}  
  
Example:  
int [] xyz = {10, 20, 30, 40};  
System.out.println(xyz[2]);//30  
------------------------------------------------  
Declaring different types of Arrays  
   
Example:  
char [] abc = {'A', 'B', 'Z'}; //Array of Characters  
int [] xyz = {10, 20, 30, 40}; //Array of Integers  
String [] a = {"UFT", "Selenium", "RFT"}; //Array of Strings  
boolean [] b ={true, false, false, true}; //Array of Boolean values  
          
System.out.println(abc[1]);//B  
System.out.println(xyz[3]);//40  
System.out.println(a[1]);//Selenium  
System.out.println(b[2]);//false  
---------------------------------------------  
c) Copy Values from one to another  
   
int [] array1 = {1, 2, 3, 4, 5};  
int array2 [] = array1;  
System.out.println(array2[2]);//3  
          
for (int i =0; i < array2.length; i++){  
System.out.println(array2[i]);  
}  
------------------------------------------  
d) Types of Arrays  
  
**Two types of Arrays**  
1) Single dimensional Array  
  
2) Multi dimensional Array  
  
**Example:**  
int [] array1 = {1, 2, 3, 4, 5};//Single dimensional Array  
int [] [] array2 = {{1, 3, 5, 7}, {2, 4, 6, 8}};// Multi dimensional Array  
          
System.out.println(array2[0][0]);//1  
System.out.println(array2[1][0]);//2  
System.out.println(array2[1][2]);//6  
--------------------------------------  
**Assignment**  
Print Multi dimensional Array (2D Array) values using Nested for loop.  
----------------------------------  
e) Advantages & Disadvantages of Arrays  
  
**Advantages:**  
Using Arrays we can optimize the code, data can be retrieved easily.  
  
We can get required data using index position  
  
**Disadvantages:**  
We can store fixed number of Elements only.  
  
It doesn't change its size during execution.  
---------------------------------------------------