Good Evening Friends....

K Prakash Babu, 12 years of exp in Java......

Welcome to Durgasoft online training.....

welcome to free workshop on java "Strings"....

Workshop on Java Strings:-
~~~~~~~~~~~~~~~~~~~~~~~~~~
01. Introduction to Strings.
02. Types of Strings Supported by Java.
03. java.lang.String Constructors
04. java.lang.String Methods Part-1
05. java.lang.String Methods Part-2 (re and format)
06. String Constant Pool (SCP Area)
07. SCP vs Heap Area
08. java.lang.StringBuffer Constructors
09. java.lang.StringBuffer Methods
10. StringBuffer vs StringBuilder
11. java.util.StringTokenizer
12. Example Programs
~~~~~~~~~~~~~~~~~~~~~~~~~~

25
26
27
28 01. Introduction to Strings.
29 ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
30 --> Collection or Sequence of Characters is called as String.
31 --> String is an Object in Java.
32 --> There is no null characters concept in java like C.
33 --> char[] to String and String to char[]
34 --> The following are the different forms of Strings.
35
36 02. Types of Strings Supported by Java.
37 ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
38
39 1. String --------> fixed length string, modifications are not allowed (immutable)
40 2. StringBuffer --> variable length string, modifications are allowed (mutable), only one thread
41 3. StringBuilder -> variable length string, modifications are allowed (mutable), mutliple thread
42 4. StringTokenizer-> it is used to divide the string into tokens, utility function.
43
44

```
Eg1:
class Test
{
        public static void main(String[] args)
        {
                String s1 = new String("welcome");
                String s2 = new String("welcome");
                System.out.println(s1==s2);//false
                System.out.println(s1.equals(s2));//true
        }
}

Eg2:
class Test
{
        public static void main(String[] args)
        {
                StringBuffer s1 = new StringBuffer("welcome");
                StringBuffer s2 = new StringBuffer("welcome");
                System.out.println(s1==s2);//false
                System.out.println(s1.equals(s2));//false
        }
}

```

```
Eg3:
class Test
{
        public static void main(String[] args)
        {
                String s1 = new String("welcome");
                StringBuffer s2 = new StringBuffer("welcome");
                System.out.println(s1.equals(s2));//false
                System.out.println(s1.contentEquals(s2));//true
        }
}


Eg4:
class Test
{
        public static void main(String[] args)
        {
                String s = new String("welcome to ");
                s.concat("java");
                System.out.println(s);//welcome to
        }
}

```

```
 93
 94 Eg5:
 95 class Test
 96 {
 97         public static void main(String[] args)
 98         {
 99                 String s = new String("welcome to ");
100                 s=s.concat("java");
101                 System.out.println(s);//welcome to java
102         }
103 }
104
105 Eg6:
106 class Test
107 {
108         public static void main(String[] args)
109         {
110                 StringBuffer s = new StringBuffer("welcome to ");
111                 s.append("java and python");
112                 System.out.println(s);//welcome to java and python
113         }
114 }
115
116 03. java.lang.String Constructors
```

```
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
1. String s = new String();
==========================
it creates an empty String object.

class Test
{
        public static void main(String[] args)
        {
                String s = new String();
                System.out.println(s);//
                System.out.println(s.length());//0
        }
}


2. String s = new String(StringLiteral);
=======================================
it creates a String obj with the given content
class Test
{
        public static void main(String[] args)
        {
                String s = new String("india");
```

```
141                 System.out.println(s);//india
142         }
143 }
144
145 3. String s = new String(char[]);
146 ================================
147 It creates a string obj with the given character array.
148
149 class Test
150 {
151         public static void main(String[] args)
152         {
153                 char[] ch= {'w','e','l','c','o','m','e'};
154                 System.out.println(new String(ch));//welcome
155         }
156 }
157
158
159 4. String s = new String(char[],int offset,int count);
160 ================================================
161 It creates a string obj with the given char[] from offset value to number of
    characters.
162
163 class Test
```

```
{
        public static void main(String[] args)
        {
                char[] ch= {'w','e','l','c','o','m','e'};
                //           0   1   2   3   4   5   6
                System.out.println(new String(ch));//welcome
                System.out.println(new String(ch,3,4));//come
                System.out.println(new String(ch,0,2));//we
                System.out.println(new String(ch,3,2));//co
                System.out.println(new String(ch,4,3));//ome
                System.out.println(new String(ch,5,2));//me
        }
}


5. String s = new String(int unicode[],int offset,int count);
==============================================================
create a string object with the given unicode values.

class Test
{
        public static void main(String[] args)
        {
                int[] uni= {'\u0041','\u0042','\u0043','\u0044','\u0045'};
```

```
188                 //                  0        1        2        3        4
189                 System.out.println(new String(uni,0,5));//ABCDE
190                 System.out.println(new String(uni,0,4));//ABCD
191                 System.out.println(new String(uni,0,3));//ABC
192                 System.out.println(new String(uni,0,2));//AB
193                 System.out.println(new String(uni,0,1));//A
194                 System.out.println(new String(uni,0,0));//
195         }
196 }
197
198 6. String s = new String(byte[]);
199 ================================
200 It creates a string for the given byte array
201
202 class Test
203 {
204         public static void main(String[] args)
205         {
206                 byte[] b = {97,98,99,100,101};
207                 System.out.println(new String(b));//abcde
208         }
209 }
210
211
```

```
7. String s = new String(byte[],int offset,int count);
==================================================
It creates a string for the given byte[] with offset value

class Test
{
        public static void main(String[] args)
        {
                byte[] b = {97,98,99,100,101};
                System.out.println(new String(b,0,5));//abcde
                System.out.println(new String(b,0,4));//abcd
                System.out.println(new String(b,0,3));//abc
                System.out.println(new String(b,0,2));//ab
                System.out.println(new String(b,0,1));//a
                System.out.println(new String(b,0,0));//
        }
}


8. String s = new String(StringBuffer);
==================================
It converts the given StringBuffer obj into String obj

Eg:
```

```
236 class Test
237 {
238         public static void main(String[] args)
239         {
240                 StringBuffer sb = new StringBuffer("java");
241                 String s = new String(sb);
242                 System.out.println(sb);//java ----> StringBuffer
243                 System.out.println(s);//java -----> String
244         }
245 }
246
247
248 9. String s = new String(StringBuilder);
249 ========================================
250 It converts the given StringBuilder obj into String obj
251 class Test
252 {
253         public static void main(String[] args)
254         {
255                 StringBuilder sb = new StringBuilder("python");
256                 String s = new String(sb);
257                 System.out.println(sb);//python ----> StringBuilder
258                 System.out.println(s);//python -----> String
259         }
```

```
260 }
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
```

284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307

308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331

332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355

356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379

380
381