**Assignments on String class**

**1. Write an application to determine the length of the String str+=“Hello world”.(using string method)**

**import** java.util.Scanner;

**public** **class** first

{

**public** **static** **void** main(String args[])

{

Scanner sc=**new** Scanner(System.***in***);

System.***out***.print("Enter the string: ");

String str = sc.nextLine();

System.***out***.println("string length is: "+str.length());

}

}

**OUTPUT:**

Enter the string: hello world

string length is: 11

**2. Write an application to join the two string “hello,” & “How are you?”(using string method).**

**import** java.util.Scanner;

**public** **class** second

{

**public** **static** **void** main(String args[])

{

String str1,str2;

Scanner sc = **new** Scanner(System.***in***);

System.***out***.print("Enter the 1st string: ");

str1=sc.nextLine();

System.***out***.print("Enter the 2nd string: ");

str2=sc.nextLine();

System.***out***.println(str1.concat(str2));

}

}

**OUTPUT:**

Enter the 1st string: Hello,

Enter the 2nd string: How are you?

Hello,How are you?

**3. Given a String "Java String pool refers to collection of Strings which are stored in heap memory", perform the following operations (Hint: all operation can be performed using String methods)**

**a. Print the string to console in lowercase**

**b. Print the string to console in uppercase**

**c. Replace all'a' character in the string with S sign**

**d. Check if the original String contains the word "collection"**

**e. Check if the following String "java string pool refers to collection of strings which are stored in heap memory" matches the original**

**f. If the string does not match check if there is another method which can be used to check if the strings are equal**

**public** **class** third

{

**public** **static** **void** main(String[] args)

{

String str=**new** String("Java String pool refers to collection of Strings which are stored in heap memory");

System.***out***.println("ORIGINAL STRING IS: " +str);

System.***out***.println();

String str1 = str.toLowerCase();

System.***out***.println("Lowercase String : " + str1);

System.***out***.println();

String str2 = str.toUpperCase();

System.***out***.println("Uppercase String : " + str2);

System.***out***.println();

String str3 = str.replace('a','s');

System.***out***.println("After Replacing all 'a' with 's' String : " + str3);

System.***out***.println();

**boolean** str4 = str.contains("collection");

System.***out***.println("Original string Contains "+"(collection)"+": Result->" + str4);

System.***out***.println();

**boolean** str5 = str.matches("collection");

System.***out***.println("Original string match with "+"(java string pool refers to collection of strings which are stored in heap memory)"+": Result->" + str5);

System.***out***.println();

**boolean** str6 = str.equals(str5);

System.***out***.println(str6+"Above string is not equals with original strings");

System.***out***.println();

}

}

**OUTPUT:**

ORIGINAL STRING IS: Java String pool refers to collection of Strings which are stored in heap memory

Lowercase String : java string pool refers to collection of strings which are stored in heap memory

Uppercase String : JAVA STRING POOL REFERS TO COLLECTION OF STRINGS WHICH ARE STORED IN HEAP MEMORY

After Replacing all 'a' with 's' String : Jsvs String pool refers to collection of Strings which sre stored in hesp memory

Original string Contains (collection): Result->true

Original string match with (java string pool refers to collection of strings which are stored in heap memory): Result->false

false Above string is not equals with original strings

------------------------------------------------------------------------------------

**Assignments on StringBuffer Class**

*Note: StringBuffer is a peer class of String that provides much of the functionality of strings. String represents fixed-length, immutable character sequences while StringBuffer represents growable and writable character sequences. StringBuffer may have characters and substrings inserted in the middle or appended to the end. It will automatically grow to make room for such additions and often has more characters preallocated than are actually needed, to allow room for growth.*

1. **Write an application to append the following strings "StringBuffer", "is a peer class of String","that provides much of", "the functionality of strings" using a StringBuffer.**

**public** **class** second\_1

{

**public** **static** **void** main(String args[])

{

String str = (**new** StringBuffer()).append("StringBuffer ").append("is a peer class of String ").append("that provides much of ").append("the functionality of strings. ").toString();

System.***out***.println(str);

}

}

**OUTPUT:** StringBuffer is a peer class of String that provides much of the functionality of strings.

**2) Insert the following string "insert text" into the string "It is used to at the specified index position at the location denoted by the sign\_**

**public** **class** second\_2

{

**public** **static** **void** main(String[] args)

{

StringBuffer sb=**new** StringBuffer("it is used to at the specified index position");

sb.insert(14 ,"insert text ");

System.***out***.println(sb);

}

}

**OUTPUT:** it is used to insert text at the specified index position

**3) Reverse the following string "This method returns the reversed object on which it was called using StringBuffer Class**

**public** **class** second\_3

{

**public** **static** **void** main(String[] args)

{

StringBuffer sb=**new** StringBuffer("This method is used to return the reverse object on which it was called");

System.***out***.println(sb.reverse());

}

}

**OUTPUT:** dellac saw ti hcihw no tcejbo esrever eht nruter ot desu si dohtem sihT

**Assignments on StringBuilder class**

*Note: StringBuilder: J2SE 5 adds a new string class to Java's already powerful string handling capabilities. This new class is called StringBuilder. It is identical to StringBuffer except for one important difference: it is not synchronized, which means that it is not thread safe. The advantage of StringBuilder is faster performance. However, in cases in which you are using multithreading, you must use StringBuffer rather than StringBuilder.*

**1) Provide solution for "Assignments on StringBuffer Class" using StringBuilder class**

1) Write an application to append the following strings "StringBuilder", "is a peer class of String","that provides much of", "the functionality of strings" using a StringBuilder.

**public** **class** third\_a

{

**public** **static** **void** main(String args[])

{

String str = (**new** StringBuilder()).append("StringBuilder ").append("is a peer class of String ").append("that provides much of ").append("the functionality of strings. ").toString();

System.***out***.println(str);

}

}

**OUTPUT:** StringBuffer is a peer class of String that provides much of the functionality of strings.

2) Insert the following string "insert text" into the string "It is used to at the specified index position at the location denoted by the sign\_

**public** **class** third\_b

{

**public** **static** **void** main(String[] args)

{

StringBuilder sb=**new** StringBuilder("it is used to at the specified index position");

sb.insert(14 ,"insert text ");

System.***out***.println(sb);

}

}

**OUTPUT:** it is used to insert text at the specified index position

3) Reverse the following string "This method returns the reversed object on which it was called using StringBuffer Class

**public** **class** third\_c

{

**public** **static** **void** main(String[] args)

{

StringBuilder sb=**new** StringBuilder("This method is used to return the reverse object on which it was called");

System.***out***.println(sb.reverse());

}

}

**OUTPUT:** dellac saw ti hcihw no tcejbo esrever eht nruter ot desu si dohtem sihT