Assignments on String Class

1) Write an application to determine the length of the String str = "Hello World". (Hint: Use String method)

**package** Sttring;

**public** **class** length {

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

{

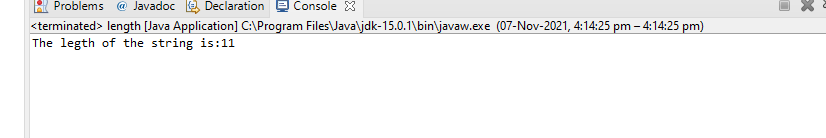
String str="Hello World";

System.***out***.println("The legth of the string is:"+str.length());

}

}

}



2) Write an application to join the two Strings "Hello," & "How are you?" (Hint: Use String method)

**package** Sttring;

**public** **class** conca {

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

{

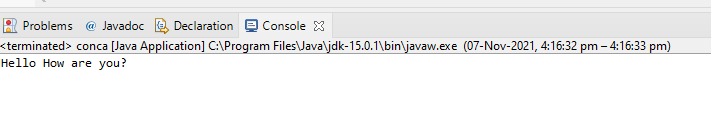
String str1="Hello ";

String str2="How are you?";

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

}

}



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)

1. Print the string to console in lowercase

b. Print the string to console in uppercase

c. Replace all'a' character in the string with $ 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

**package** Sttring;

**public** **class** Third {

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

{

String str1="java String pool refers collection of strings which are stored in heap memory";

System.***out***.println(str1.toUpperCase());

System.***out***.println(str1.toLowerCase());

System.***out***.println(str1.replace('a', '$'));

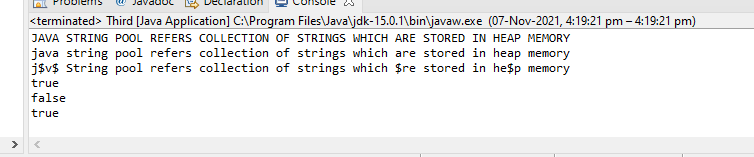
System.***out***.println(str1.contains("collection"));

System.***out***.println(str1.equals("java string pool refers collection of strings which are stored in heap memory"));

System.***out***.println(str1.equalsIgnoreCase("java string pool refers collection of strings which are stored in heap memory"));

}

}



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.

**package** Sttring;

**public** **class** four {

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

StringBuffer sb = **new** StringBuffer("StringBuffer");

sb.append("is a peer class of string");

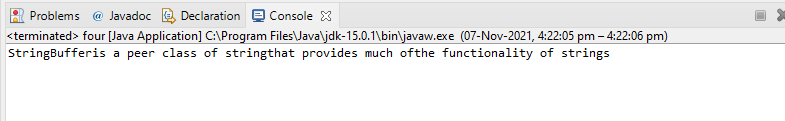
sb.append("that provides much of");

sb.append("the functionality of strings");

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

}

}



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

**package** Sttring;

**public** **class** Five {

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

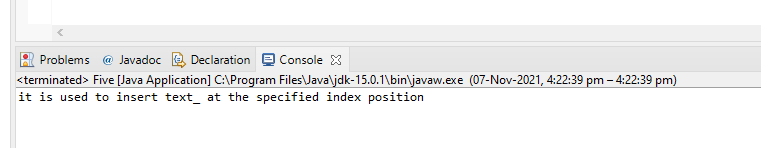
StringBuffer in = **new** StringBuffer("it is used to \_ at the specified index position");

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

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

}

}



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

**package** Sttring;

**public** **class** Six {

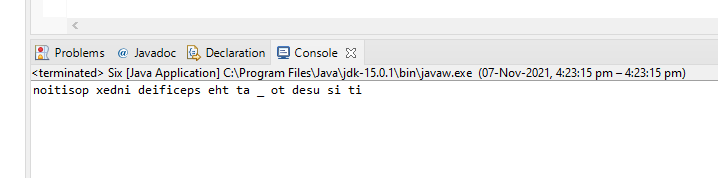
**public** **static** **void** main(String[] args) {

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

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

}

}



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

**package** Sttring;

**public** **class** Seven {

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

StringBuilder in = **new** StringBuilder("StringBuffer");

in.append("is a peer class of string");

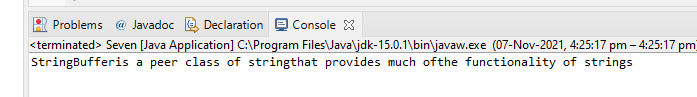
in.append("that provides much of");

in.append("the functionality of strings");

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

}

}



**package** Sttring;

**public** **class** eight {

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

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

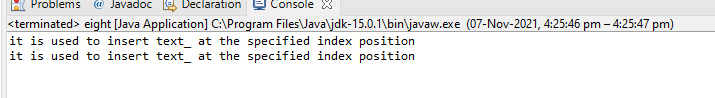
in.insert(14,"insert text");

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

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

}

}



**package** Sttring;

**public** **class** nine {

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

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

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

}

}

