Assignment on String Class

1. Write an application to determine the length of the string str =”Hello world”

**public** **class** Length {

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

String str="Hello World";

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

}}

Output: 11

1. Write an application to join the two string” Hello” & “How are You”

**public** **class** Concat {

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

String str1="Hello ";

String str2="How are you";

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

}

}

Output: Hello How are you

1. Given a String “Java String pool refers to collection of strings which are stored in heap memory”, perform the following operations
2. Print the sting to console in lowercase

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

String str1="Java String Pool Refers To Collection Of Strings Which Are Stored In Heap Memory ";

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

}

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

1. Print the string to console in uppercase

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

String str1="Java String Pool Refers To Collection Of Strings Which Are Stored In Heap Memory ";

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

}

}

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

1. Replace all ‘a’ character in the string with $ sign

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

{

String s1="Java string pool refers to collection of strings which are in heap memory";

String replaceString=s1.replace("a", "$");

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

}

Output: J$v$ string pool refers to collection of strings which $re in he$p memory

1. Check if the original contains the world” collection”

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

{

String txt="java string pool refer to collection of strings which are in heap memory";

String str="collection";

**boolean** result=txt.contains(str);

**if**(result)

{

System.***out***.println(str + " "+ "is present in the string");

}

**else**

{

System.***out***.println(str + " "+ "is not present in the string");

}

}

Output: collection is present in the string

1. Check if the following string ="Java String Pool Refers To Collection Of Strings Which Are Stored In Heap Memory " matches the original.

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

String str1="Java String Pool refers to Collection of Strings which are stored in heap memory ";

String str2="java string pool refers to collection of strings which are stored in heap memory ";

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

}

}

Output: false

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

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

{

String s1,s3;

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

System.***out***.println("enter the string:");

s1=sc.nextLine();

System.***out***.println("Enter the string to be checked:");

s3=sc.nextLine();

**boolean** s2=s1.equals(s1);

System.***out***.println("Print:"+s2);

}

Output:

enter the string:

java string pool refer to collection of strings which are in heap memory

Enter the string to be checked:

heap

Print: true

Assignments on StringBuffer Class

1. Write an applicatipn to append the following string “StringBuffer”, “is a peer class of String”,”that provides much of”,” the functionality of strings” using a StringBuffer.

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

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

sb.append( " is a peer of String"+" that provides much of "+" the functionality of strings");

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

}

Output:

StringBuffer is a peer of String that provides much of the functionality of strings

1. 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** **static** **void** main(String[] args) {

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

sb.insert(13, " insert text");

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

}

Output: It is used to insert text at the specified index position

1. Reverse the following string “ this method returns the reversed object on which it was called” using StringBuffer class.

Ans🡪

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

StringBuffer sb= **new** StringBuffer("this method returns the reversed object on which it was called");

sb.reverse();

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

}

Output: dellac saw ti hcihw no tcejbo desrever eht snruter dohtem siht

Assignment on StringBuilder Class

1. Reverse the following string “ this method returns the reversed object on which it was called” using StringBuffer class.

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

StringBuilder sb= **new** StringBuilder("this method returns the reversed object on which it was called");

sb.reverse();

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

}

Output: dellac saw ti hcihw no tcejbo desrever eht snruter dohtem siht

1. Write an applicatipn to append the following string “StringBuffer”, “is a peer class of String”,”that provides much of”,” the functionality of strings” using a StringBuffer.

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

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

sb.append( " is a peer of String"+" that provides much of "+" the functionality of strings");

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

}

Output: StringBuilder is a peer of String that provides much of the functionality of strings

1. 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** **static** **void** main(String[] args) {

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

sb.insert(13, " insert text");

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

}

Output:

It is used to insert text at the specified index position