Assignment on String Class

1. Write an application to determine the length of the String str=”Hello World.(Hint Use Stringmethod)

**package** org.string.app;

**public** **class** String1

{

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

{

String str="Hello world";

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

}

}

OUTPUT:

string length is 11

1. Write an application to join the two Strings “Hello,”&”How are you?”(Hint:Use String method)

**package** org.string.app;

**public** **class** JoinTwoStrings

{

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

{

String s1=**new** String("Hello");

String s2=**new** String("How are you?");

String s=String.*join*(" ",s1,s2);

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

}

}

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(Hint: all operation can be performed usingString methods)
2. Print the String to console in lowercase.

**package** org.string.app;

**public** **class** Stringlowercase

{

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

{

String str="JAVA STRING POOL REFERS TO COLLECTION OF STRINGS WHICH ARE STORE IN HEAP MEMORY";

String result=str.toLowerCase();

System.***out***.println("Original String:" +" "+str);

System.***out***.println("Lower case String:" +" "+result);

}

}

OUTPUT:

Original String: JAVA STRING POOL REFERS TO COLLECTION OF STRINGS WHICH ARE STORE IN HEAP MEMORY

Lower case String: java string pool refers to collection of strings which are store in heap memory

1. Print the string to console in uppercase.

**package** org.string.app;

**public** **class** Stringuppercase

{

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

{

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

String result=str.toUpperCase();

System.***out***.println("Original String:" +" "+str);

System.***out***.println("Upper case String:" +" "+result);

}

}

OUTPUT:

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

Upper case String: JAVA STRING POOL REFERS TO COLLECTION OF STRINGS WHICH ARE IN HEAP MEMORY

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

**package** org.string.app;

**public** **class** ReplaceString

{

**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 orginal String contains the word “Collection”.

**package** org.string.app;

**public** **class** OriginalString

{

**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.

**package** org.string.app;

**public** **class** MatchString

{

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

{

String Str1= **new** String("java string pool refers to collection of string which are stored in heap memory");

String Str2 = **new** String("java String pool refers to collection of String which are stored in heap memory");

System.***out***.print("Checking if java String pool refers to collection of String which are stored in heap memory( case sensitive ) : ");

System.***out***.println( Str1.regionMatches(6, Str2,0, 41));

}

}

OUTPUT:

Checking if java String pool refers to collection of String which are stored in heap memory( case sensitive ) : 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.

**package** org.string.app;

**import** java.util.Scanner;

**public** **class** StringEqual

{

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

{

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

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

String s1=sc.nextLine();

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

String s2=sc.nextLine();

**boolean** s3=s1.equals(s2);

System.***out***.println("Print:"+s3);

}

}

OUTPUT:

enter the string:

Manisha

Enter the string to be checked:

Manu

Print:false

Assignment 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 SrtingBuffer represents growable and writable 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** org.string.app;

**public** **class** Append

{

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

{

StringBuffer sb=**new** StringBuffer("StringBuffer is a peer class of a string");

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

sb.append(" " +" the fuctionalities of string");

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

}

}

OUTPUT:

StringBuffer is a peer class of a string that provides much of the fuctionalities of string

1. Insert the following string “insert text” into the string “it is used to\_at the specified index postion” at the loction denoted by the Sign.

**package** org.string.app;

**public** **class** InsertText

{

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

{

StringBuffer sb=**new** StringBuffer("It is used to insert text 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 an which it was called” using StringBuffer Class.

**package** org.string.app;

**public** **class** Reverse

{

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

{

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

sb.reverse();

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

}

}

OUTPUT:

dellac saw ti hcihw no tcejbo esrever eht snruter dohtem sihT

Assignment on StringBulider Class

Note: StringBulider : J2SE 5 adds a new string class to ‘java’s already powerful string handing capabilities. Thid new class is called StringBulider. It is identical to StingBuffer except for one important difference : it is not synchronized, Which means that it is not threads safe. The advantage of StringBulider is a faster performance. However, in cases in which you are using mutitherading, you must use StringBuffer rather than StringBulider.

Provide solution for “Assignmets on StringBuffer Class” using StringBulider class

1. Write an application to append to append the following strings “StringBulider”,

“is a peer of a String”, “that provides much of”, “the funcationalities of String” using StringBulider.

**package** org.string.app;

**public** **class** AppendStringBulider

{

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

{

StringBuilder sb=**new** StringBuilder("StringBuilder is a peer class of a string");

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

sb.append(" " +" the functionalities of string");

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

}

}

OUTPUT:

StringBuilder is a peer class of a string that provides much of the functionalities of string

1. Insert the following string “ insert text” into the string “it is used to\_at the specified

Index position” at the location denoated by\_sign using StringBulider.

**package** org.string.app;

**public** **class** InsertStringBulider

{

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

{

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

sb.insert(11, "insert text ");

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

}

}

OUTPUT:

It is used insert text to at the specified index position

1. Reverse the following string :This method is used to return the reverse object on which it was called” using StringBulider class.

**package** org.string.app;

**public** **class** ReverseStringBuilder

{

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

{

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

sb.reverse();

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

}

}

OUTPUT:

dellac saw ti hcihw no tcejbo esrever eht snruter dohtem sihT