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

**public** **class** strlength {

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

{

String str="Hello wrold";

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

}

}

Output:

string length is 11

**2.Write an application to join the two string “Hello” and “How are you?”**

**public** **class** strjoin {

**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 2:

Hello How are you?

**3. Given a string “java string pool refers to collection of string which are store in heap memory” perform the following operations:**

a. print the string to console in lower case

**public** **class** lowercase {

**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

**b. Print the string to console in upper case**

**public** **class** uppercase {

**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

**c. Replace all ‘a’ character in a string with ‘$’ sign**

**public** **class** replace {

**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

**d. Check if the original string contains the word “collection”**

**public** **class** contain {

**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

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

**public** **class** StringMatch

{

**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

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

**import** java.util.Scanner;

**public** **class** stringequal {

**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

**STRINGBUFFER**

**1.Write an application to append the following strings “StringBuffer”, “is a peer of a String”, “that provides much of”, “the functionalities of string” using StringBuffer.**

**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

**2. Insert the following string “insert text” into the string “it is used to \_at the specified index position” at the location denoted by \_sign using StringBuffer.**

**public** **class** insert {

**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

3.Reverse the following string “This method is used to return the reverse object on which it was called” using StringBuffer class.

**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

**STRINGBUILDER**

**String Builder is not thread safe, we use StringBuilder because it is more efficient.**

**Provide solution for “Assignments of stringBuffer class” using StringBuilder”**

**1. Write an application to append the following strings “StringBuilder”, “is a peer of a String”, “that provides much of”, “the functionalities of string” using StringBuilder.**

**public** **class** append {

**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

**2. Insert the following string “insert text” into the string “it is used to \_at the specified index position” at the location denoted by \_sign using StringBuilder.**

**public** **class** insert {

**public** **class** insert {

**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

**3. Reverse the following string “This method is used to return the reverse object on which it was called” using StringBuilder class.**

**public** **class** reverse {

**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