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

public class StrLength{

public static void main(String[] args)

{

String str;

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

Scanner sc=new Scanner(System.in);

str=sc.nextLine();

System.out.println("Length of the string:"+ str.length());

}

}

Output:

Enter the string:

Hello world

Length of the string: 11

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

public class StrConcat {

public static void main(String[] args) {

Scanner sc= new Scanner(System. in);

System.out.println("Enter first String : ");

String str1 = sc.nextLine();

System.out.print("Enter second string : ");

String str2 = sc.nextLine();

String str=str1.concat(str2);

System.out.println("Display Concatenated String: \n "+str);

}

}

Output

Enter first string:

Hello

Enter second string: How are you??

Display Concatenated String:

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 pool {

public static void main(String[] args)

{

String s1, s2;

Scanner sc=new Scanner(System.in);

System.out.println("Uppercase letter:");

s1= sc.nextLine();

s2 = s1.toLowerCase();

System.out.println("lowercase:" +s2);

}

}

Uppercase letter:

SIRI

lowercase: siri

b. Print the string to console in upper case

public class pool {

public static void main(String[] args)

{

String s1, s2;

Scanner sc=new Scanner(System.in);

System.out.println("Lowercase letter:");

s1=sc.nextLine();

s2=s1.toUpperCase();

System.out.println("uppercase:" +s2);

}

}

Output:

Lowercase letter:

hello world

uppercase: HELLO WORLD

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

public class pool {

public static void main(String[] args)

{

String s1, s2;

Scanner sc=new Scanner(System.in);

System.out.println("Enter the string:");

s1=sc.nextLine();

s2 = s1.replace("a","$");

System.out.println("string:" +s2);

}

}

Output:

Enter the string:

an apple

string: $n $pple

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

public class pool {

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.contains(s1);

System.out.println("string:" +s2);

}

}

Output:

Enter the string:

Java uses oops concept

Enter the string to be checked:

oops

string: true

Enter the string:

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

Enter the string to be checked:

collection

string: true

e. use another method to check strings are equal.

public class pool {

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:

collection

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 Strbuffer {

public static void main(String[] args){

StringBuffer sb=new StringBuffer();

sb.append("Hi");

sb.append(“Hello”);

System.out.println(sb);

}

}

Output:

Hi Hello

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 Strbuffer {

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 is used to return the reverse object on which it was called” using StringBuffer class

public class Strbuffer {

public static void main(String[] args)

{

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

sb.reverse();

System.out.println(sb);

}

}

Output:

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

String Builder

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 buff {

public static void main(String[] args)

{

StringBuilder sb=new StringBuilder("StringBuilder, is a peer of a String");

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

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

System.out.println(sb);

}

}

Output:

StringBuilder, is a peer 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 denoted by \_sign using StringBuilder.

public class buff {

public static void main(String[] args)

{

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

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

System.out.println(sb);

}

}

Output

it is used to index text 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 StringBuilder class.

public class buff {

public static void main(String[] args)

{

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

sb.reverse();

System.out.println(sb);

}

}

Output:

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