

Assignments on String Class

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

```
# Armstrong.java Armstrongnum... CompoundInt... Subjects.java TaxAmount.java Login.java SearchArray... Bubblesort.java Student.java Bank.java Length.java x
1 package Assignment3.java;
2 import java.util.*;
3 public class Length {
4     public static void main(String args[]) {
5         Scanner sc = new Scanner(System.in);
6         String h = sc.next();
7         int k = h.length();
8         System.out.println("Length is "+k);
9     }
10 }
11
```

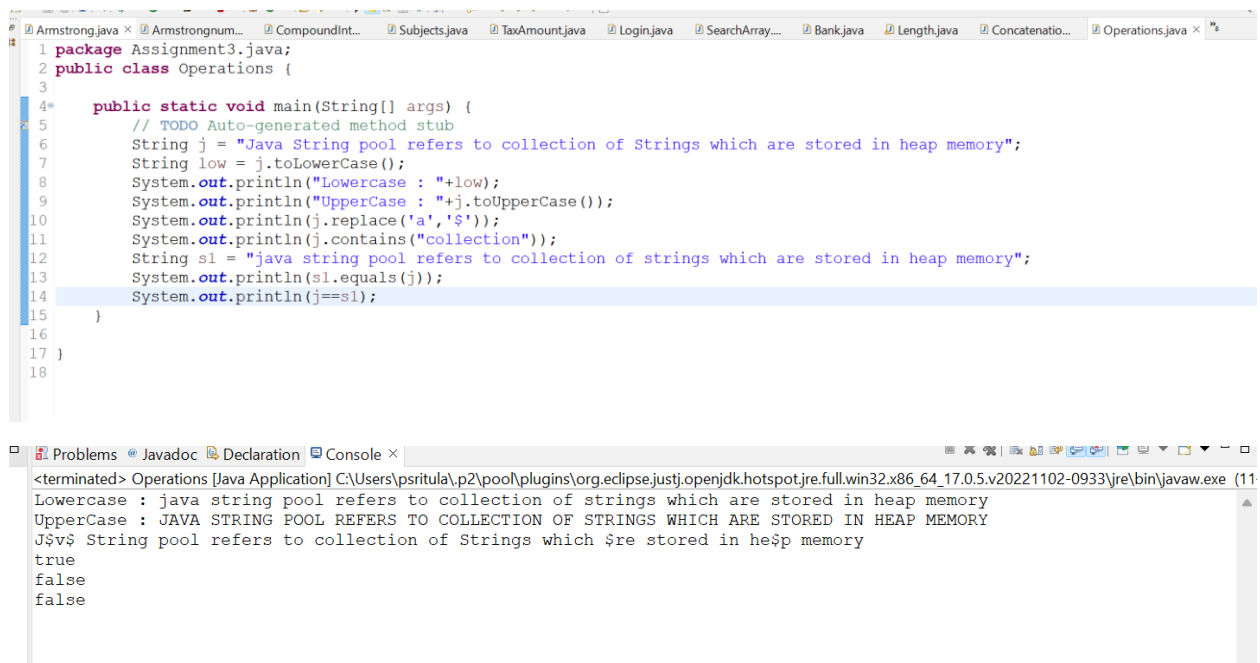
```
Problems Javadoc Declaration Console
<terminated> Length [Java Application] C:\Users\psritula\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.5.v20221102-0933\jre\bin\javaw.exe (11-Dec-2022, 3:28:11 pm - 3:28:29 pm) [pid: 19976]
Likhitha
Length is 8
```

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

```
# Armstrong.java Armstrongnum... CompoundInt... Subjects.java TaxAmount.java Login.java SearchArray... Bubblesort.java Bank.java Length.java Concatenatio... x
1 package Assignment3.java;
2
3 public class Concatenation {
4
5     public static void main(String[] args) {
6         String s1 = "Hello & ";
7         String s2 = s1.concat("How are you");
8         System.out.println(s2);
9         // TODO Auto-generated method stub
10
11     }
12
13 }
14
```

```
Problems Javadoc Declaration Console
<terminated> Concatenation [Java Application] C:\Users\psritula\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.5.v20221102-0933\jre\bin\javaw.exe
Hello & How are you
```

- 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)
- Print the string to console in lowercase
 - Print the string to console in uppercase
 - Replace all ‘a’ character in the string with \$ sign
 - Check if the original String contains the word “collection”
 - Check if the following String “java string pool refers to collection of strings which are stored in heap memory” matches the original
 - If the string does not match check if there is another method which can be used to check if the strings are equal



The screenshot shows an IDE with a Java file named `Operations.java` and its console output. The code in `Operations.java` is as follows:

```
1 package Assignment3.java;
2 public class Operations {
3
4     public static void main(String[] args) {
5         // TODO Auto-generated method stub
6         String j = "Java String pool refers to collection of Strings which are stored in heap memory";
7         String low = j.toLowerCase();
8         System.out.println("Lowercase : " + low);
9         System.out.println("UpperCase : " + j.toUpperCase());
10        System.out.println(j.replace('a', '$'));
11        System.out.println(j.contains("collection"));
12        String s1 = "java string pool refers to collection of strings which are stored in heap memory";
13        System.out.println(s1.equals(j));
14        System.out.println(j==s1);
15    }
16 }
17 }
18 }
```

The console output shows the results of the operations:

```
<terminated> Operations [Java Application] C:\Users\psritula\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.5.v20221102-0933\jre\bin\javaw.exe (11
Lowercase : java string pool refers to collection of strings which are stored in heap memory
UpperCase : JAVA STRING POOL REFERS TO COLLECTION OF STRINGS WHICH ARE STORED IN HEAP MEMORY
J$v$v String pool refers to collection of Strings which $re stored in he$p memory
true
false
false
```

Assignments 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 `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.

- 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`.

```
1 package Assignment3.java;
2
3 public class StrBuf {
4
5     public static void main(String[] args) {
6         StringBuffer b = new StringBuffer();
7         b.append("StringBuffer");
8         b.append("is a peer class of String");
9         b.append("that provides much of");
10        b.append("the functionality of Strings");
11        System.out.println(b);
12        // TODO Auto-generated method stub
13    }
14 }
15
16 }
17
```

Problems Javadoc Declaration Console ×

<terminated> StrBuf [Java Application] C:\Users\psritula\p2\pool\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64_17.0.5.v20221102-0933\jre\bin\javaw.exe (11-StringBuffer is a peer class of String that provides much of the functionality of Strings

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 _

```
1 package Assignment3.java;
2
3 public class Index {
4
5     public static void main(String[] args) {
6         // TODO Auto-generated method stub
7         String s2="It is used to _ at the specified index position";
8         StringBuffer b1 = new StringBuffer(s2);
9         System.out.println(b1.insert(14, "INSERT TEXT"));
10    }
11 }
12 }
13
```

Problems Javadoc Declaration Console ×

<terminated> Index [Java Application] C:\Users\psritula\p2\pool\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64_17.0.5.v20221102-0933\jre\bin\javaw.exe (11-Dec-It is used to INSERT TEXT_ at the specified index position

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

```
1 package Assignment3.java;
2
3 public class Reverse {
4
5     public static void main(String[] args) {
6         // TODO Auto-generated method stub
7         String str = "This method returns the reversed object on which it was called";
8         StringBuffer b = new StringBuffer(str);
9         System.out.println(b.reverse());
10    }
11
12 }
13
```

Problems Javadoc Declaration Console ×

<terminated> Reverse [Java Application] C:\Users\psritula\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.5.v20221102-0933\jre\bin\javaw.exe (1
dellac saw ti hcihw no tcejbo desrever eht snruter dohtem sihT

Assignments on StringBuffer Class

Note: StringBuilder: J2SE 5 adds a new string class to Java's already powerful string handling capabilities. This new class is called **StringBuilder**. It is identical to **StringBuffer** except for one important difference: it is not synchronized, which means that it is not thread safe. The advantage of **StringBuilder** is faster performance. However, in cases in which you are using multithreading, you must use **StringBuffer** rather than **StringBuilder**.

- 1) Provide solution for "Assignments on StringBuffer Class" using **StringBuilder** class

```
1 package Assignment3.java;
2
3 public class Builder {
4
5     public static void main(String[] args) {
6         // TODO Auto-generated method stub
7         String st="String Builder";
8         StringBuilder st1 = new StringBuilder(st);
9         System.out.println(st1.reverse());
10
11     }
12
13 }
14
```

Problems Javadoc Declaration Console ×

<terminated> Builder [Java Application] C:\Users\psritula\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.5.v20221102-0933\jre\bin\javaw.exe (11-Dec-2022 11:05:00 AM)
redliuB gnirtS