|  |  |
| --- | --- |
| *A close up of a logo  Description automatically generated* | *DEPARTMENT OF COMPUTER ENGINEERING* |

|  |  |
| --- | --- |
| Semester | S.E. Semester III – Computer Engineering |
| Subject | Object Oriented Programming Using Java (Skill Based Lab) |
| Subject Professor In-charge | Prof. Indu Anoop |
| Laboratory | Online Lab |

|  |  |  |
| --- | --- | --- |
| Student Name | Trisha Shah | |
| Roll Number | 20102A0004 | |
| Grade and Subject Teacher’s Signature |  |  |

|  |  |  |
| --- | --- | --- |
| Experiment | 6B | |
| Problem Statement | WAP in java to demonstrate String Buffer | |
| Resources / Apparatus Required | Hardware: Computer System | Software: jdk 1.8, Eclipse / Notepad++/IntelliJ IDEA |
| Details | The primitive data types are specified by the keywords int, double, etc. While these keywords start with a lowercase letter, the keyword String, which represents the string data type, starts with an uppercase letter. This is because of the fact that keyword String is the name of a predefined class. A Java string is an instantiated object of the String class. A String variable is simply a variable that stores a reference to an object of the class String. You declare a String variable in much the same way as you define a variable of one of the basic types. You can also three way to initialize it in the declaration.  Major difference between string function and string buffer   1. Once we create a string object we can’t perform any changes in the existing object. If we are trying to perform any changes with those changes a new object will be created. This non changeable nature is called immutability of the string object 2. Once we create a StringBuffer object we can perform any type of changes in the existing object. This changeable nature is called mutability of the StringBuffer object. | |
| Code | public class String\_StringBuffer {  public static void main(String[] args) {  //String class is immutable    String s= new String("Trisha");  s.concat("Shah");  System.out.println(s);    //String Buffer class is mutable  StringBuffer sb= new StringBuffer("Trisha");  sb.append("Shah");  System.out.println(sb);    //Learning concept of SCP(String Constant Pool) for String Class  String a1=new String("Java");  String a2=new String("Java");  System.out.println(a1==a2);//both references are pointing to different object-->false  System.out.println(a1.equals(a2));//content of both objects is the same -->true (String equals mthod overrrides the Object parent Class's euals methos to have its's own implementation of content comparison      //Learning concept of String Buffer ,Note: No SCP concept in String Buffer  StringBuffer sb1=new StringBuffer("Java");  StringBuffer sb2=new StringBuffer("Java");  System.out.println(sb1==sb2);//both references are pointing to different object-->false  System.out.println(sb1.equals(sb2));//the StringBuffer Class doesnot override the equals method of the Parent Object Class due to which it retains the fucntionality of Object Class Equals method which is used for reference comparison  }  } | |
| Output |  | |
| Conclusion | Thus, we have successfully executed a program to explore different string buffer functions | |