***Dt : 9/9/2022***

***faq:***

***define charAt() method?***

***=>charAt() method is used to retrieve character from the String***

***based on index value.***

***syntax:***

***char ch = str.char(index);***

***faq:***

***define length() method?***

***=>length() method is used to find the length of String.***

***syntax:***

***int len = str.length();***

***==========================================================***

***faq:***

***define String Concatenation process?***

***=>The process of combining multiple Strings into a single String***

***is known as String Concatenation process.***

***=>Concatenation process in Java canbe done in two ways:***

***(i)Using concat() method***

***(ii)Uisng '+' symbol***

***Ex : DemoString5.java***

***package maccess;***

***public class DemoString5 {***

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

***String s1 = "java";***

***String s2 = "language";***

***String s3 = "program";***

***String s4 = s1.concat(s3);***

***System.out.println(s4.toString());***

***String s5 = s1.concat(s2).concat(s3);***

***System.out.println(s5.toString());***

***String s6 = s1+s2+s3;***

***System.out.println(s6.toString());***

***}***

***}***

***o/p:***

***javaprogram***

***javalanguageprogram***

***javalanguageprogram***

***Note:***

***=>In String Concatenation process separate object is created to***

***hold concatenated Strings,because the Strings objects are Immutable***

***objects.***

***=======================================================***

***faq:***

***define String Comparision process?***

***=>The process of comparing two strings is known as String***

***Comparision process.***

***=>String Comparision process canbe done in three ways:***

***(a)Using 'equals()' method***

***(b)Using 'compareTo()' method***

***(c)Uisng 'is equal to'(==) operator***

***(a)Using 'equals()' method:***

***=>'equals()' method will compare two Strings and generate boolean***

***result.***

***Method Signature:***

***public boolean equals(java.lang.Object);***

***public boolean equalsIgnoreCase(java.lang.String);***

***Note:***

***=>In realtime equals() method is used for User Authentication***

***process.***

***(b)Using 'compareTo()' method:***

***=>'compareTo()' method is also used to compare two Strings and***

***generate int result.***

***Method Signature:***

***public int compareTo(java.lang.String);***

***public int compareToIgnoreCase(java.lang.String);***

***syntax:***

***int z = s1.compareTo(s2);***

***if z==0 then Strings are equal***

***if z>0 then s1>s2***

***if z<0 then s1<s2***

***Note:***

***=>In realtime compareTo() method is used in Sorting process.***

***=============================================================***

***Ex\_Program:***

***wap to read Student rollNo and branch,perform validation and***

***verification to check the rollNo belongs to branch or not?***

***01 - CIVIL***

***02 - EEE***

***03 - MECH***

***04 - ECE***

***05 - CSE***

***GenerateBranch.java***

***package test;***

***public class GenerateBranch {***

***public String generate(String code){***

***return switch(code) {***

***case "01":yield "CIVIL";***

***case "02":yield "EEE";***

***case "03":yield "MECH";***

***case "04":yield "ECE";***

***case "05":yield "CSE";***

***default:yield null;***

***};***

***}***

***}***

***DemoString6.java(MainClass)***

***package maccess;***

***import java.util.\*;***

***import test.GenerateBranch;***

***public class DemoString6 {***

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

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

***System.out.println("Enter the RollNO:");***

***String rollNo = s.nextLine();***

***int len = rollNo.length();***

***if(len==10)***

***{***

***String code = rollNo.substring(6, 8);***

***GenerateBranch gb = new GenerateBranch();***

***String br = gb.generate(code);***

***if(br==null)***

***{***

***System.out.println("Invalid branch code...");***

***}//end of if***

***else***

***{***

***System.out.println("Enter the stuName:");***

***String name = s.nextLine();***

***System.out.println***

***("Enter the branch:(CIVIL/EEE/MECH/ECE/CSE)");***

***String branch = s.nextLine().toUpperCase();***

***if(br.equals(branch))***

***{***

***System.out.println("RollNo matched with branch");***

***System.out.println("RollNo:"+rollNo);***

***System.out.println("Name:"+name);***

***System.out.println("Branch:"+branch);***

***}//end of if***

***else***

***{***

***System.out.println("RollNo Not-matched with branch");***

***}***

***}***

***}//end of if***

***else***

***{***

***System.out.println("Invalid RollNO...");***

***}***

***s.close();***

***}***

***}***

***o/p:***

***Enter the RollNO:***

***1234560290***

***Enter the stuName:***

***Raj***

***Enter the branch:(CIVIL/EEE/MECH/ECE/CSE)***

***eee***

***RollNo matched with branch***

***RollNo:1234560290***

***Name:Raj***

***Branch:EEE***

***=========================================================***

***Assignment:***

***Update above program by reading 6 sub marks and calculating result.***

***============================================================***

***Dt : 12/9/2022***

***Ex-program : DemoString7.java***

***package maccess;***

***import java.util.\*;***

***public class DemoString7 {***

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

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

***System.out.println("Enter the String-1");***

***String s1 = s.nextLine().trim();***

***System.out.println("Enter the String-2");***

***String s2 = s.nextLine().trim();***

***System.out.println("=====equals()=====");***

***boolean k = s1.equalsIgnoreCase(s2);***

***if(k) {***

***System.out.println("Strings are equal....");***

***}else{***

***System.out.println("Strings are Not-equal...");***

***}***

***System.out.println("=====compareTo()=====");***

***int z = s1.compareToIgnoreCase(s2);***

***if(z==0) {***

***System.out.println("Strings are equal...");***

***}else {***

***System.out.println("Strings are Not-equal...");***

***}***

***s.close();***

***}***

***}***

***o/p:***

***Enter the String-1***

***nit***

***Enter the String-2***

***nit***

***=====equals()=====***

***Strings are equal....***

***=====compareTo()=====***

***Strings are equal...***

***==========================================================***

***Note:***

***=>'IgnoreCase' specify to compare the content without considering***

***the case.***

***=>'trim()' method is used to remove the spaces before and after***

***the String.***

***==========================================================***

***(c)Uisng 'is equal to'(==) operator:***

***=>'is eqaul to'(==) operator will compare the object references,***

***which means it will compare the content of an objects.***

***Ex : DemoString8.java***

***package maccess;***

***public class DemoString8 {***

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

***//String Literal process***

***String s1 = "nit";***

***String s2 = "nit";***

***//new operator process***

***String s3 = new String("hyd");***

***String s4 = new String("hyd");***

***System.out.println("====String Literal process====");***

***if(s1==s2) {***

***System.out.println("String are equal...");***

***}else {***

***System.out.println("Strings are Not-equal...");***

***}***

***System.out.println("====new operator process====");***

***if(s3==s4) {***

***System.out.println("String are equal...");***

***}else {***

***System.out.println("Strings are Not-equal...");;***

***}***

***}***

***}***

***o/p:***

***====String Literal process====***

***String are equal...***

***====new operator process====***

***Strings are Not-equal...***

***============================================================***

***faq:***

***define String Constant pool?***

***=>The separate partition of HeapArea where String objects are***

***created is known as String Constant pool.***

***Advantage:***

***=>In String Constant Pool only Unique String objects are created,***

***which means it will not allow duplicate String object creation.***

***==============================================================***

***faq:***

***wt is the diff b/w 'String Constant pool' and 'new operator process'***

***in creating String objects?***

***(i)String Literal process:***

***=>In String Literal process,the execution control will check***

***the String Constant pool is any object having same data,***

***=>If Object not available then new Object is created.***

***=>If Object already available,then without creating new object***

***uses the reference of existing object.***

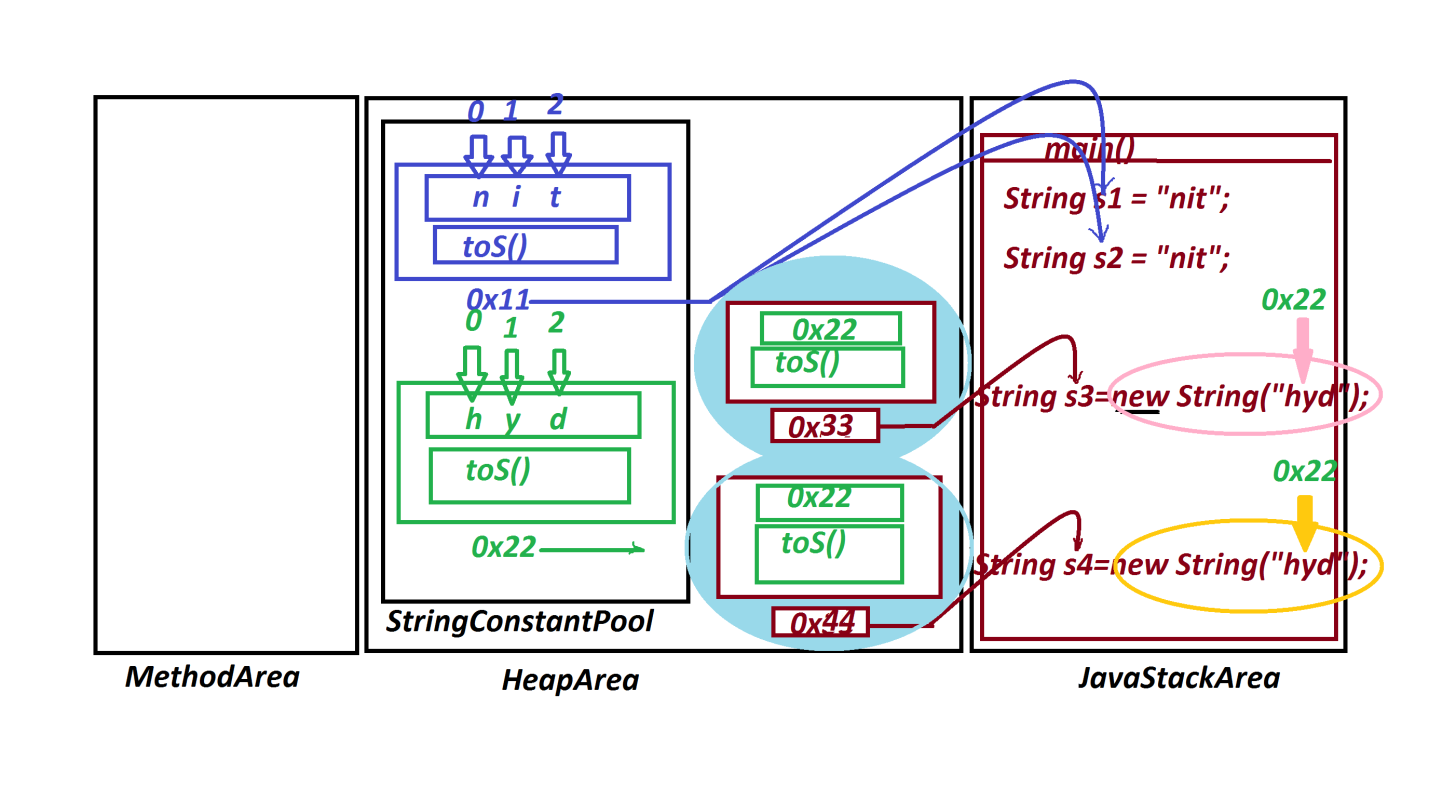
***(ii)new operator process:***

***=>In new operator process one object is created directly in the***

***HeapArea,but the object will hold the reference of object created***

***in String Constant pool.***

***Diagram:***

******

***===================================================***

***\*imp***

***2.java.lang.StringBuffer class:***

***=>The objects which are created using 'java.lang.StringBuffer'***

***are mutable objects.(The objects once created can be modified are***

***known as Mutable objects)***

***=>The following are four constructors from 'StringBuffer':***

***public java.lang.StringBuffer();***

***public java.lang.StringBuffer(int);***

***public java.lang.StringBuffer(java.lang.String);***

***public java.lang.StringBuffer(java.lang.CharSequence);***