1. Write a program to concatenate StringBuilder & StringBuffer objects.

**class** Assignment {

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

// **TODO** Auto-generated method stub

StringBuilder sbl1=**new** StringBuilder("abc");

StringBuilder sbl2=**new** StringBuilder("xyz");

StringBuilder sbl3=sbl1.append(sbl2);

System.***out***.println(sbl3.toString());

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

StringBuffer sbf1=**new** StringBuffer();

sbf1.append("Welcome to ");

StringBuffer sbf2=**new** StringBuffer();

sbf2.append("Java");

StringBuffer sbf3=sbf1.append(sbf2);

System.***out***.println(sbf3.toString());

}

}

1. Write a Program to get substring of a StringBuffer.

**public** **class** Assignment{

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

// **TODO** Auto-generated method stub

StringBuffer sbf = **new** StringBuffer("Stringbuffer");

System.out.println("String: "+sbf);

String subString= sbf.substring(3);

System.out.println("String starting from index 3= "+subString);

}

}

1. Write a program to display the length and capacity of String, StringBuilder and StringBuffer.

**class** Assignment{

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

// **TODO** Auto-generated method stub

String stobj1=**new** String("Bibek");

String stobj2=**new** String("Basak");

String resString=stobj1 + stobj2;

System.out.println(resString);

System.out.println("Length: " + resString.length());

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

StringBuilder stbobj1=**new** StringBuilder("Bibek");

StringBuilder stbobj2=**new** StringBuilder("Basak");

StringBuilder s=stbobj1.append(stobj2);

System.out.println(s.toString());

System.out.println("Capacity: " +s.capacity());

System.out.println("Length: " + s.length());

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

StringBuffer sbf1=**new** StringBuffer();

sbf1.append("Bibek");

StringBuffer sbf2=**new** StringBuffer();

sbf2.append("Basak");

StringBuffer sbf=sbf1.append(sbf2);

System.out.println(sbf.toString());

System.out.println("Capacity: " +sbf.capacity());

System.out.println("Length: " + sbf.length());

}

}

1. Write a program to check whether two given strings contains same set of characters as well as having same length.

**import** java.util.Arrays;

**public** **class** Assignment {

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

// **TODO** Auto-generated method stub

String str1 = "madam";

String str2 = "ammad";

**char**[] chars1 = str1.toCharArray();

**char**[] chars2 = str2.toCharArray();

Arrays.sort(chars1);

Arrays.sort(chars2);

**if**(Arrays.equals(chars1,chars2)) {

System.out.println(str1 + " and " + str2 + " have same set of character");

} **else** {

System.out.println(str1 + " and " + str2 + " not have same set of character");

}

System.out.println("Length of str1: " + str1.length());

System.out.println("Length of str2: " + str2.length());

**if**(str1.length()==str2.length()) {

System.out.println(str1 + " and " + str2 + " have same length");

}

**if**(str2.length()!=str2.length()) {

System.out.println(str1 + " and " + str2 + " not have the same length");

}

}

}

1. Write a program to lexicographically arrange the given strings "Raman" , "Aman" , "Vikram" , "Shyam" and "Bhuvan".

**public** **class** Assignment {

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

String[] name = { "Chiku","Rohit","Rahul","Asish","Rajesh"};

**int** n = 5;

System.out.println("Before Sorting");

**for**(**int** i = 0; i < n; i++) {

System.out.println(name[i]);

}

**for**(**int** i = 0; i < n-1; ++i) {

**for** (**int** j = i + 1; j < n; ++j) {

**if** (name[i].compareTo(name[j]) > 0) {

String temp = name[i];

name[i] = name[j];

name[j] = temp;

}

}

}

System.out.println("\nAfter performing lexicographical order: ");

**for**(**int** i = 0; i < n; i++) {

System.out.println(name[i]);

}

}

}

1. Wrapper Classes (Integer, Byte, Short, Long, Float, Double, Character, Boolean)

//Create objects of all the wrapper classes and print then on console, with using constructor.

**public** **class** Assignment {

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

// **TODO** Auto-generated method stub

Integer i = 10;

Double d = 11.65;

Character c = 'T';

Boolean b= **true**;

System.out.println(i);

System.out.println(d);

System.out.println(c);

System.out.println(b);

}

}\*/

1. Write a program to demonstrate boxing and un-boxing.

**public** **class** Assignment {

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

//boxing

Integer objInteger = Integer.valueOf(10);

//un-boxing

**int** a = objInteger.intValue();

System.out.println(a);

}

}

//8.Write a program to demonstrate autoboxing and unboxing.

**public** **class** Assignment {

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

// **TODO** Auto-generated method stub

//autoboxing

Integer objInteger = 10;

//un-boxing

**int** a = objInteger;

System.out.println(a);

}

}

|  |
| --- |
|  |
|  |

9.Create a list of 10 Integer objects and process the list element by element to count objects containing even value. Print the total number of odd objects after counting even objects.

**package** core\_Java;

**import** java.util.ArrayList;

**public** **class** Solution5 {

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

ArrayList<Integer> integers = **new** ArrayList<>();

integers.add(20);

Integer obj = Integer.*valueOf*(43);

integers.add(obj);

integers.add(60);

integers.add(77);

integers.add(50);

integers.add(30);

integers.add(250);

integers.add(3);

integers.add(137);

integers.add(10);

System.***out***.println("printing list");

System.***out***.println(integers);

**int** evenCount = 0;

**for** (Integer integer : integers) {

**if**(integer.intValue() % 2 ==0){

evenCount++;

}

}

**int** oddCount = integers.size() - evenCount;

System.***out***.println("odd count = " + oddCount);

}}

10- Create a List of 10 Student objects, where Student is a user defined class. Student class contains three private fields int rollNo, String name, and double cpi.

|  |
| --- |
|  |
| StudentclassmustoverrideStringtoString()methodofObjectclass.Kindlysortthestudentbythere name. And also delete the students whose name starts with "A" & "J". |
|  |

**package** core\_Java;

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

**class** Student {

**private** **int** rollno;

String name;

**private** **double** cpi;

**public** Student(**int** rollno, String name, **double** cpi)

{

**this**.rollno = rollno;

**this**.name = name;

**this**.cpi = cpi;

}

**public** **int** getRollno() {

**return** rollno;

}

**public** **void** setRollno(**int** rollno) {

**this**.rollno = rollno;

}

**public** String getName() {

**return** name;

}

**public** **void** setName(String name) {

**this**.name = name;

}

**public** **double** getCpi() {

**return** cpi;

}

**public** **void** setCpi(**double** cpi) {

**this**.cpi = cpi;

}

**public** String toString()

{

**return** **this**.rollno+" "+**this**.name+" "+**this**.cpi+" ";

}

}

**class** Sortbynames **implements** Comparator<Student>

{

**public** **int** compare(Student a, Student b){

**return** a.name.compareTo(b.name);

}

}

**class** Student1{

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

Student s = **new** Student(111,"G",1234);

List<Student> std = **new** ArrayList<Student>();

std.add(**new** Student(123,"A",23456));

std.add(**new** Student(122,"B",23457));

std.add(**new** Student(103,"Y",23458));

std.add(**new** Student(113,"D",23459));

std.add(**new** Student(223,"Q",23451));

std.add(**new** Student(493,"W",23452));

std.add(**new** Student(23,"E",23454));

std.add(**new** Student(223,"P",23466));

std.add(**new** Student(146,"G",22456));

std.add(**new** Student(128,"X",21456));

System.***out***.println("Unsorted ");

**for** (**int** i=0;i<std.size() ;i++ ) {

System.***out***.println(std.get(i));

}

Collections.*sort*(std, **new** Sortbynames());

System.***out***.println("\n Sorted by Student Name ");

**for** (**int** i=0;i<std.size() ;i++ ) {

System.***out***.println(std.get(i));

}

**for** (**int** i=0;i<std.size()-9;i++) {

System.***out***.println(std);

}

}

}

11- Create a List & Set of 10 integers and demonstrate ordered and unordered collection as well as duplicate allowed and not allowed. Demonstrate addAll() method also.

**package** core\_Java;

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

**class** List\_Test

{

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

List<Integer> int\_list = **new** ArrayList<Integer>();

int\_list.add(23);

int\_list.add(45);

int\_list.add(37);

int\_list.add(890);

int\_list.add(37);

int\_list.add(234);

int\_list.add(178);

int\_list.add(65);

int\_list.add(345);

int\_list.add(890);

System.***out***.println("Array List : "+int\_list.toString());

Set<Integer> int\_set = **new** HashSet<Integer>();

int\_set.add(23);

int\_set.add(45);

int\_set.add(67);

int\_set.add(890);

int\_set.add(67);

int\_set.add(234);

int\_set.add(178);

int\_set.add(45);

int\_set.add(345);

int\_set.add(10);

System.***out***.println("Hash Set : "+int\_set.toString());

List<Integer> int\_list1 = Collections.***EMPTY\_LIST***;

Collections.*addAll*(int\_list1 = **new** ArrayList<Integer>(), 23,34,44,56,78,98,65);

System.***out***.println("Array List : "+int\_list1.toString());

Set<Integer> int\_set1 = **new** HashSet<Integer>();

int\_set1.add(35);

int\_set1.add(90);

int\_set1.addAll(int\_set);

System.***out***.println("HashSet : "+int\_set1.toString());

}

}

12- Create a List of 10 Integer objects and try to access 15th index. Properly analysis the output.

**package** core\_Java;

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

**class** List\_Test2

{

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

List<Integer> int\_list2 = Collections.***EMPTY\_LIST***;

Collections.*addAll*(int\_list2 = **new** ArrayList<Integer>(), 23,34,44,56,78,98,65,567,145,75);

System.***out***.println("Array List : "+int\_list2.toString());

**int** element = int\_list2.get(15);

System.***out***.println("The element at 15 index is "+element);

}

}