**Q1. Write a program to replace a substring inside a string with another string ?**

**Solution**

**import java.util.Scanner;**

***/\*Question-1\*/***

**public class ReplaceSubstring {**

***//will replace all occurrences of toBeReplaced***

**static String replaceSubstringWithString(String originalString,String toBeReplaced,String newString)**

**{**

**return originalString.replace(toBeReplaced,newString);**

**}**

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

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

**System.*out*.println("Enter a string");**

**String originalString=in.next();**

**System.*out*.println("Enter substring to be replaced");**

**String tobeReplaced=in.next();**

**System.*out*.println("Enter new substring");**

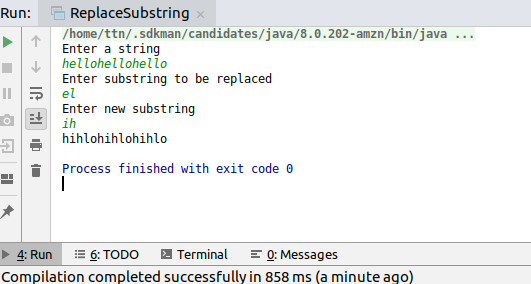
**String newString=in.next();**

**System.*out*.println(*replaceSubstringWithString*(originalString,tobeReplaced,newString));**

**}**

**}**

**Output**



**Q2. Write a program to find the number of occurrences of the duplicate words in a string and print them ?**

**Solution**

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

***/\*Question--2\*/***

**public class DuplicateWords {**

**static HashMap<String,Integer> findDuplicateOccurrences(String inputString)**

**{**

**HashMap<String,Integer>wordOccurrenceMap=new HashMap<String, Integer>();**

***//converting a string to a single case so that occurrence of a single word can be counted correctly, irrespective of case***

**StringTokenizer wordsinString=new StringTokenizer(inputString.toLowerCase());**

**while(wordsinString.hasMoreTokens())**

**{**

**String currentWord=wordsinString.nextToken();**

**if(!wordOccurrenceMap.containsKey(currentWord))**

**{**

**wordOccurrenceMap.put(currentWord,1);**

**}**

**else**

**{**

**wordOccurrenceMap.put(currentWord,wordOccurrenceMap.get(currentWord)+1);**

**}**

**}**

**return wordOccurrenceMap;**

**}**

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

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

**System.*out*.println("Enter a string");**

**String inputString=in.nextLine();**

**HashMap<String,Integer>wordOccurrenceMap=*findDuplicateOccurrences*(inputString);**

**System.*out*.println("word-----------------occurrence");**

**for(Map.Entry<String,Integer>curPair:wordOccurrenceMap.entrySet())**

**{**

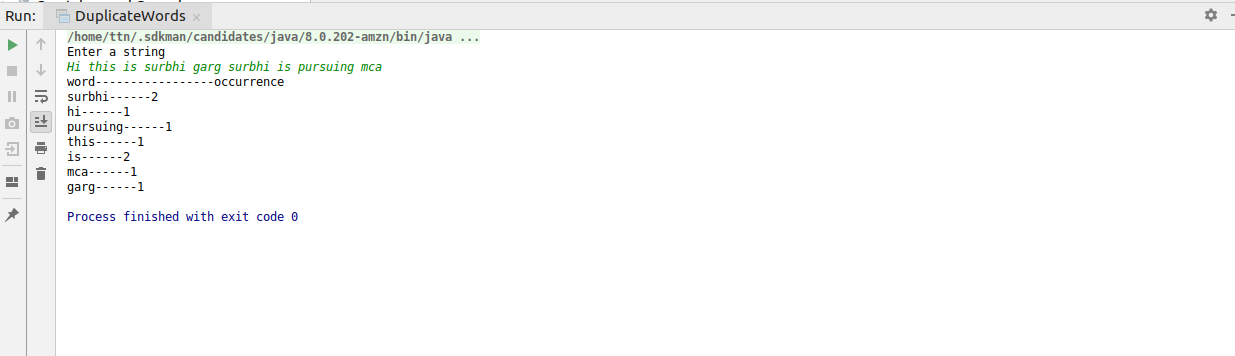
**System.*out*.println(curPair.getKey()+"------"+curPair.getValue());**

**}**

**}**

**}**

**Output**



**Q3. Write a program to find the number of occurrences of a character in a string without using loop?**

**Solution**

**import java.util.Arrays;**

**import java.util.Scanner;**

**import java.util.StringTokenizer;**

***/\*Question-3\*/***

**public class FindFrequency {**

***//way-1***

**static int findFrequencyOfAChar(String inputString,char searchKey)**

**{**

**int length=inputString.length();**

**inputString= inputString.toLowerCase();**

***//total length-length of string without that character***

**int charCount=length-inputString.replaceAll(Character.*toString*(Character.*toLowerCase*(searchKey)),"").length();**

**return charCount;**

**}**

***//way-2***

**static int findOccurrencyOfAChar2(String inputString,char searchKey)**

**{**

***//split will cause occurrence+1 splits.***

**return inputString.toLowerCase().split(String.*valueOf*(Character.*toLowerCase*(searchKey)),-1).length-1;**

**}**

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

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

**System.*out*.println("Enter a string");**

**String inputString=in.next();**

**System.*out*.println("Enter character to be searched");**

**char searchKey=in.next().charAt(0);**

**int charCount=*findFrequencyOfAChar*(inputString,searchKey);**

**System.*out*.println("Result By way1");**

**System.*out*.println(searchKey+" occurs "+charCount+" times");**

**System.*out*.println("Result By way2");**

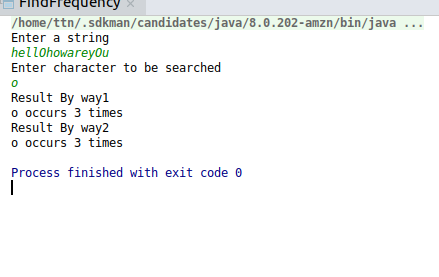
**charCount=*findOccurrencyOfAChar2*(inputString,searchKey);**

**System.*out*.println(searchKey+" occurs "+charCount+" times");**

**}**

**}**

**Output**



**Q4. Calculate the number & Percentage Of Lowercase Letters,Uppercase Letters, Digits And Other Special Characters In A String**

**Solution**

***/\* Question-4\*/***

**import java.util.HashMap;**

**import java.util.Map;**

**import java.util.Scanner;**

**public class FindCountAndPercentage {**

**static HashMap<String,Integer>countNumberOfOccurrencesBySymbolType(String input) {**

**HashMap<String, Integer> typeCountmap = new HashMap<String, Integer>();**

**String digit="digit";**

**String lowercase="lowercase";**

**String uppercase="uppercase";**

**String specialchar="specialchar";**

**for(int i=0;i<input.length();i++)**

**{**

**char curChar=input.charAt(i);**

**if(Character.*isLowerCase*(curChar))**

**{**

***//if key lowercase is not found put 0, else increment by1***

**typeCountmap.put(lowercase,typeCountmap.get(lowercase)==null?1:typeCountmap.get(lowercase)+1);**

**}**

**else if(Character.*isUpperCase*(curChar))**

**{**

***//if key uppercase is not found put 0, else increment by1***

**typeCountmap.put(uppercase,typeCountmap.get(uppercase)==null?1:typeCountmap.get(uppercase)+1);**

**}**

**else if(Character.*isDigit*(curChar))**

**{**

***//if key digit is not found put 0, else increment by1***

**typeCountmap.put(digit,typeCountmap.get(digit)==null?1:typeCountmap.get(digit)+1);**

**}**

**else**

**{**

**typeCountmap.put(specialchar,typeCountmap.get(specialchar)==null?1:typeCountmap.get(specialchar)+1);**

**}**

**}**

**return typeCountmap;**

**}**

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

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

**System.*out*.println("Enter a string");**

**String input=in.next();**

**HashMap<String,Integer> typeCountMap=*countNumberOfOccurrencesBySymbolType*(input);**

**System.*out*.println("Symbol type--------count------percentage");**

**for(Map.Entry<String,Integer>curEntry:typeCountMap.entrySet())**

**{**

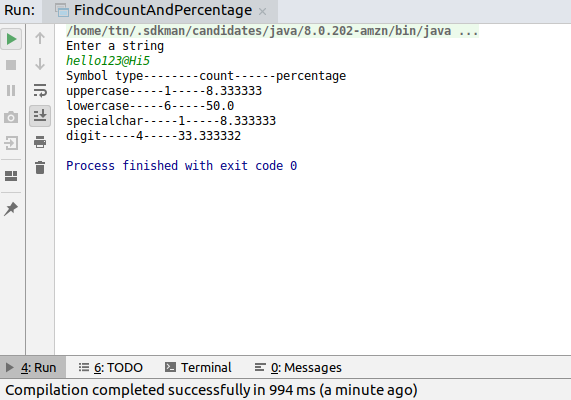
**System.*out*.println(curEntry.getKey()+"-----"+curEntry.getValue()+"-----"+((float)curEntry.getValue()\*100)/input.length());**

**}**

**}**

**}**

**Output**



**Q5. Find common elements between two arrays.**

**Solution**

**import java.util.ArrayList;**

**import java.util.Scanner;**

***/\*Question--5\*/***

**public class Intersect {**

**static void readArray(int[]arr,int arraySize)**

**{**

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

**for(int index=0;index<arraySize;index++)**

**{**

**System.*out*.println(index+"th element:");**

**arr[index]=in.nextInt();**

**}**

**}**

**static void findCommonElements(int[]firstArray, int[]secondArray, ArrayList<Integer>intersectingElements)**

**{**

**int count=0;**

**for(int curFirst:firstArray)**

**{**

**for(int curSecond:secondArray)**

**{**

**if(curFirst==curSecond)**

**intersectingElements.add(curFirst);**

**}**

**}**

**}**

***//This can be optimized using merge sort logic, provided the arrays are sorted***

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

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

**System.*out*.println("Enter the number of elements of first array:");**

**int firstArraySize=in.nextInt();**

**System.*out*.println("Enter the number of elements of second array:");**

**int secondArraySize=in.nextInt();**

**int[]firstArray=new int[firstArraySize];**

**int[]secondArray=new int[secondArraySize];**

**System.*out*.println("Enter the elements of first array");**

***readArray*(firstArray,firstArraySize);**

**System.*out*.println("Enter the elements of second array");**

***readArray*(secondArray,secondArraySize);**

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

***findCommonElements*(firstArray,secondArray,intersectingElements);**

**System.*out*.println("Common elements between entered arrays are:");**

**for(int ele:intersectingElements)**

**{**

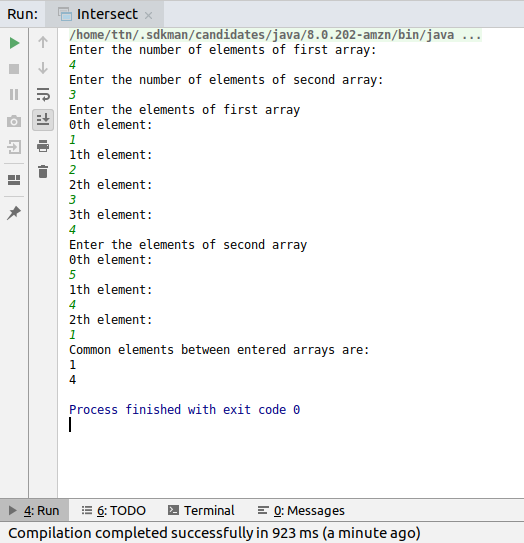
**System.*out*.println(ele);**

**}**

**}**

**}**

**Output**



**Q6. There is an array with every element repeated twice except one. Find that element**

**Solution**

**import java.util.Scanner;**

***//Question-6***

**public class FindNonRepeating {**

***//Find the first occurrence of an element in array.***

**static int findFirst(int[]inputArr,int searchKey)**

**{**

**for(int i=0;i<inputArr.length;i++)**

**{**

**if(inputArr[i]==searchKey)**

**return i;**

**}**

**return -1;**

**}**

***//Find last occurrence of an element in array.***

**static int findLast(int[]inputArr,int searchKey)**

**{**

**int first=*findFirst*(inputArr,searchKey);**

**if(first==-1)**

**return -1;**

**if(first==inputArr.length-1)**

**return inputArr.length-1;**

**for(int i=first+1;i<inputArr.length;i++)**

**{**

**if(inputArr[i]==searchKey)**

**return i;**

**}**

**return first;**

**}**

**static int findNonRepeatingElement(int[]inputArray)**

**{**

**for(int index=0;index<inputArray.length;index++)**

**{**

**int firstIndex=*findFirst*(inputArray,inputArray[index]);**

**int lastIndex=*findLast*(inputArray,inputArray[index]);**

***/\*\*If there are chances that element will not be found in array, we can code it here,***

***\* however,it will not be a case here.***

***\*/***

**if(firstIndex==-1)**

**continue;**

***/\*If element occurs only once, it will have same first and last index\*/***

**if(lastIndex-firstIndex==0) {**

**return index;**

**}**

**}**

**return -1;**

**}**

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

**int foundNonRepeating=0;**

**int[]inputArray;**

**int eleCount;**

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

**System.*out*.println("Enter number of elements of array");**

**eleCount=in.nextInt();**

**inputArray=new int[eleCount];**

**System.*out*.println("Enter elements of array:");**

**for(int i=0;i<eleCount;i++)**

**{**

**System.*out*.println(i+1 +"th element:");**

**inputArray[i]=in.nextInt();**

**}**

**foundNonRepeating=*findNonRepeatingElement*(inputArray);**

**if(foundNonRepeating==-1)**

**{**

**System.*out*.println("There is no distinct element present in array.");**

**}**

**else**

**{**

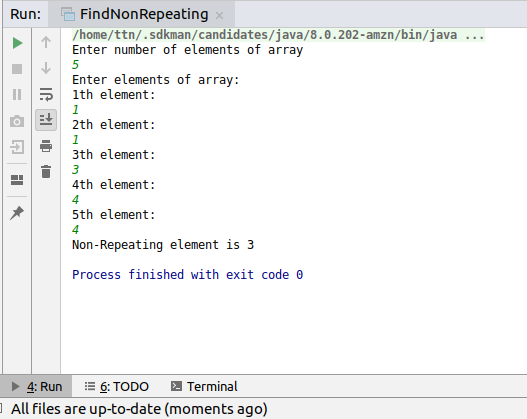
**System.*out*.println("Non-Repeating element is "+inputArray[foundNonRepeating]);**

**}**

**}**

**}**

**Output**



**Q7. Write a program to print your Firstname,LastName & age using static block,static method & static variable respectively**

**Solution**

**public class** StaticUse {

**static** String *firstName*=**"Surbhi"**;

**static** String *lastName*=**"Garg"**;

**static int** *age*=23;

**static** {

System.***out***.println(**"Inside static"**);

System.***out***.println(**"Surbhi Garg 23"**);

}

**static void** display()

{

System.***out***.println(**"In display"**);

System.***out***.println(**"Surbhi Garg 23"**);

}

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

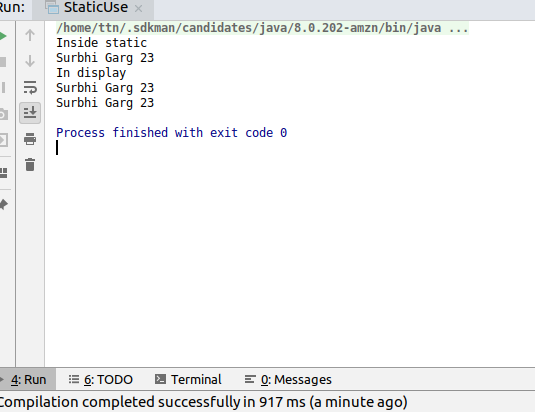
*display*();

System.***out***.println(*firstName*+**" "**+*lastName*+**" "**+*age*);

}

}

**Output**



**Q8. Write a program to reverse a string and remove character from index 4 to index 9 from the reversed string using String Buffer**

**Solution**

**import java.util.Scanner;**

**public class ReverseAndRemove {**

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

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

**System.*out*.println("Enter a string");**

**StringBuffer input=new StringBuffer(in.next());**

**int length=input.length();**

***//using delete in this case won't throw an error, will simply remove characters found in given range***

**if(length-1<9){**

**System.*out*.println("Complete operation can't be performed");**

**}**

**else**

**{**

***//end is exclusive, that's why taken 10***

**input.reverse().delete(4,10);**

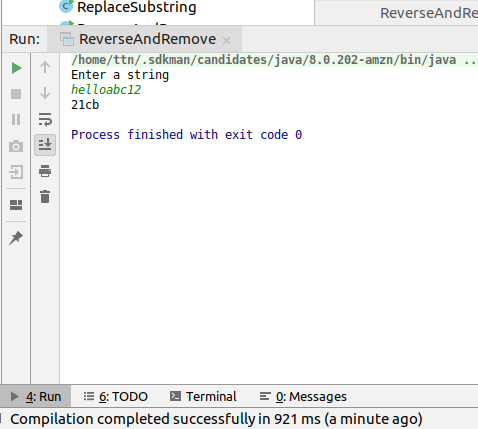
**System.*out*.println(input);**

**}**

**}**

**}**

**Output**



**Q9.Write a program to display values of enums using a constructor & getPrice() method (Example display house & their prices)**

**Solution**

**public class EnumConstructorAndMethod {**

**enum House{**

***HOUSE1*(5000000),*HOUSE2*(10000000),*HOUSE3*(4000000);**

**private long price;**

**House(long price) {**

**this.price = price;**

**}**

**long getPrice() {**

**return price;**

**}**

**}**

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

**System.*out*.println("House--------Cost");**

**for(House curHouse:House.*values*())**

**{**

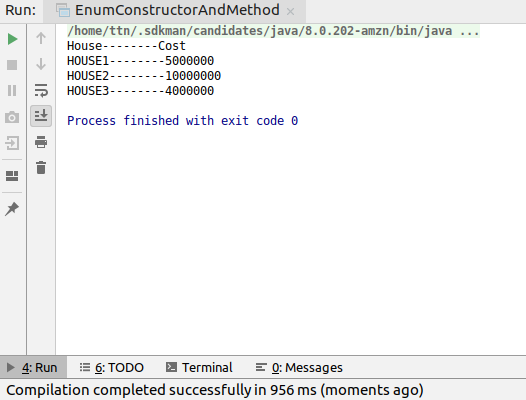
**System.*out*.println(curHouse+"--------"+curHouse.getPrice());**

**}**

**}**

**}**

**Output**



**Q10.Write a single program for following operation using overloading**

**A) Adding 2 integer number**

**B) Adding 2 double**

**C) multiplying 2 float**

**D) multiplying 2 int**

**E) concate 2 string**

**F) Concate 3 String**

**Solution**

***/\*Question-10\*/***

**public class Overloading {**

**static int add(int num1,int num2)**

**{**

**return num1+num2;**

**}**

**static double add(double num1,double num2)**

**{**

**return num1+num2;**

**}**

**static float multiply(float num1,float num2)**

**{**

**return num1\*num2;**

**}**

**static int multiply(int num1,int num2)**

**{**

**return num1\*num2;**

**}**

**static String concatenate(String first,String second)**

**{**

**return first+second;**

**}**

**static String concatenate(String first,String second,String third)**

**{**

**return first+second+third;**

**}**

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

**System.*out*.println(*add*(12,14));**

**System.*out*.println(*add*(12.5,14.5));**

**System.*out*.println(*add*(11.0,14.0));**

**System.*out*.println(*multiply*(2,4));**

**System.*out*.println(*multiply*(2.5f,4.5f));**

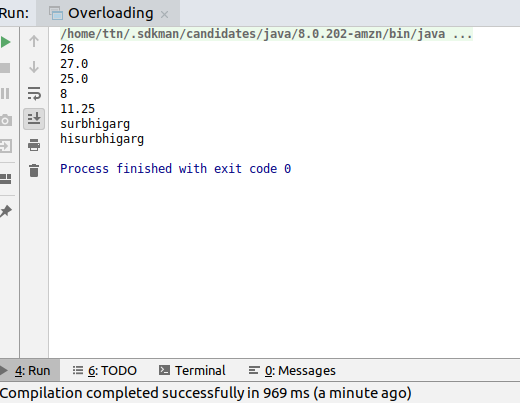
**System.*out*.println(*concatenate*("surbhi","garg"));**

**System.*out*.println(*concatenate*("hi","surbhi","garg"));**

**}**

**}**

**Output**



**Q11.Create 3 sub class of bank SBI,BOI,ICICI all 4 should have method called getDetails which provide there specific details like rateofinterest etc,print details of every banks**

**Solution**

**package bank;**

**public class Bank {**

**float *rateOfInterest*;**

**int *numberOfCustomers*;**

**int *numberOfBranches*;**

**public Bank(float rateOfInterest,int numberOfCustomers,int numberOfBranches)**

**{**

**this.*rateOfInterest*=rateOfInterest;**

**this.*numberOfCustomers*=numberOfCustomers;**

**this.*numberOfBranches*=numberOfBranches;**

**}**

**void getDetails()**

**{**

**System.*out*.println("Interest rate: "+*rateOfInterest*+"Number Of Customers: "+*numberOfCustomers*+"Number Of Branches"+*numberOfBranches*);**

**}**

**}**

**package bank;**

**public class Boi extends Bank {**

**public Boi(float rateOfInterest,int numberOfCustomers,int numberOfBranches)**

**{**

**super(rateOfInterest,numberOfCustomers,numberOfBranches);**

**}**

**}**

**package bank;**

**public class Icic extends Bank{**

**public Icic(float rateOfInterest,int numberOfCustomers,int numberOfBranches)**

**{**

**super(rateOfInterest,numberOfCustomers,numberOfBranches);**

**}**

**}**

**package bank;**

**public class Sbi extends Bank {**

**public Sbi(float rateOfInterest,int numberOfCustomers,int numberOfBranches)**

**{**

**super(rateOfInterest,numberOfCustomers,numberOfBranches);**

**}**

**}**

**Output**

