**Institute of Engineering & Management**

**Department of Computer Science & Engineering**

**Object Oriented Programming (IT) Labfor 3rd year 5th semester 2018**

**Code: CS594D**

**Date:** 18/10/18

**WEEK-11**

**Assignment-1**

**Problem Statement:** Write a java code to detect the mth character of a given string.

**Source code:**

classTeststring

{

public static void main(String args[])

{

String str="Programming" ;

System.out.println("string = " + str);

intlen=str.length();

System.out.println(len);

chara\_char = str.charAt(3);

System.out.println("character at index 3 is : "+a\_char);

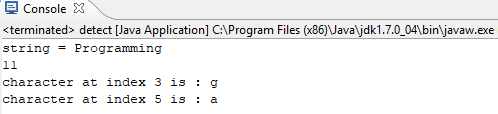
charar\_char = str.charAt(5);

System.out.println("character at index 5 is : "+ar\_char);

}

}

**Screen-Shot:**



**Assignment-2**

**Problem Statement:** Write a java program to sort n number of strings in lexicographic order.

**Source code:**

**import** java.util.Scanner;

**public** **class** detect

{

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

{

**int** n;

String temp;

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

System.*out*.print("Enter number of names you want to enter:");

n = s.nextInt();

String names[] = **new** String[n];

Scanner s1 = **new** Scanner(System.*in*);

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

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

names[i] = s1.nextLine();

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

{

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

{

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

{

temp = names[i];

names[i] = names[j];

names[j] = temp;

}

}

}

System.*out*.println("Names in Sorted Order:");

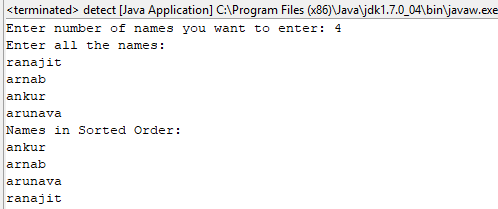
**for** (**int** i = 0; i < n ; i++)

System.*out*.println(names[i]);

}

}

**Screen-Shot:**

****

**Assignment-3**

**Problem Statement:** Write a JAVA program to generate a histogram of a given string .

**Source code:**

**import** java.util.Scanner;

**public** **class** histogram

{

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

{

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

**final** **int** LETTERS\_IN\_ALPHABET = 26;

**int**[] letterCounter = **new** **int**[LETTERS\_IN\_ALPHABET];

System.*out*.print("Enter string: ");

String string = kb.nextLine();

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

{

**char** letterThere = string.charAt(i);

**int** placeInLetterCtr = *whereInLetterCtr*(letterThere);

letterCounter[placeInLetterCtr]++;

}

*printNumbers*(letterCounter);

*printLetters*();

}

**public** **static** **int** whereInLetterCtr(**char** letter)

{

**int** i =0 ;

**for**(**char** comparisonLetter = 'a'; comparisonLetter<= 'z'; comparisonLetter++)

{

**if**(letter == comparisonLetter)

{

**return** i;

}

i++;

}

**return** i;

}

**public** **static** **void** printNumbers(**int**[] array)

{

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

{

System.*out*.print(" "+array[i]);

}

System.*out*.println();

}

**public** **static** **void** printLetters()

{

**for**(**char** letter ='a'; letter <='z'; letter++)

{

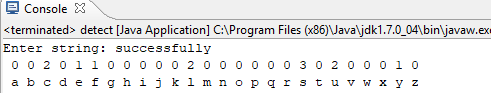
System.*out*.print(" "+letter);

}

}

}

**Screen-Shot:**



**Assignment-4**

**Problem Statement:** Write a java program to print odd and even numbers using two separate threads (First thread is created by Thread class and the Second thread is created by Runnable interface).

**Source code:**

**class** Odd **extends** Thread

{

**public** **void** run()

{

**for**(**int** i=1;i<10;i=i+2)

{

System.*out*.println(i);

**try** {Thread.*sleep*(1000);}

**catch**(Exception e){}

}

}

}

**class** Even **implements** Runnable

{

**public** **void** run()

{

**for**(**int** i=2;i<=10;i=i+2)

{

System.*out*.println(i);

**try** {Thread.*sleep*(1000);}

**catch**(Exception e){}

}

}

}

**class** detect

{

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

{

Odd obj1 = **new** Odd();

Runnable obj2= **new** Even();

Thread t = **new** Thread(obj2);

obj1.start();

**try** {Thread.*sleep*(10);}

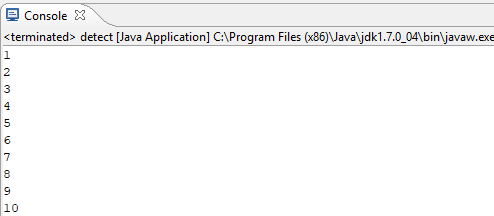
**catch**(Exception e){}

t.start();

}

}

**Screen-Shot:**



**Assignment-5**

**Problem Statement:** write a java code to demonstrate the use of the following methods.

1. currentThread()
2. getname()
3. setname()

**Source code:**

**class** Test **extends** Thread

{

**public** **void** run()

{

System.*out*.println("Thread started\nCurrent thread = " + Thread.*currentThread*());

}

}

**class** detect

{

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

{

Test obj1 = **new** Test();

Test obj2 = **new** Test();

Test obj3 = **new** Test();

System.*out*.println("Name of thread1 = " + obj1.getName());

System.*out*.println("Name of thread2 = " + obj2.getName());

System.*out*.println("Name of thread3 = " + obj3.getName());

System.*out*.println("\nAfter setting name,the name of the threads are :");

obj1.setName("Music");

obj2.setName("Cinema");

obj3.setName("Game");

System.*out*.println("Name of thread1 = " + obj1.getName());

System.*out*.println("Name of thread2 = " + obj2.getName());

System.*out*.println("Name of thread3 = " + obj3.getName());

System.*out*.print("\n");

obj1.start();

obj2.start();

obj3.start();

}

}

**Screen-Shot:**

