

Programme Name: **BCS**

Course Code**: CSC 1510**

Course Name: **Programming Fundamentals**

Assignment: **Third**

Date of Submission: **5/10/2020**

**Submitted by: Submitted to:**

Student Name: **Dipesh Tha Shrestha** Name: **Prakash Chandra**

Semester: **Second**

Section: **A**

Intake**: 2019 September**

**QUESTION NO 1**

1. **Explain static variable**.

*Answer* **Static variable** is variable which belongs to the class and initialized only once at the start of the execution. It is a variable which belongs to the class and not to object(instance ). Static variables are initialized only once, at the start of the execution. These variables will be initialized first, before the initialization of any instance variables.

* A single copy to be shared by all instances of the class
* A static variable can be accessed directly by the class name and doesn’t need any object

Syntax :

<***class-name>.<variable-name>***

1. **Write a java program to count the number of objects created, using the concept of static variable**.

=The java program to count the number of object created using the concept of static variable is given below:

class Test {

    static int noOfObjects = 0;

    // Instead of performing increment in the constructor

    // instance block is preferred to make this program generic.

    {

        noOfObjects += 1;

    }

    // various types of constructors

    // that can create objects

    public Test()

    {

    }

    public Test(int n)

    {

    }

    public Test(String s)

    {

    }

    public static void main(String args[])

    {

        Test t1 = new Test();

        Test t2 = new Test(5);

        Test t3 = new Test("GFG");

        // We can also write t1.noOfObjects or

        // t2.noOfObjects or t3.noOfObjects

        System.out.println(Test.noOfObjects);

    }

}

**Output:**

3

**Question number 2**

**You’ve been provided a .csv (comma separated values) file of student IDs and test scores. You are to write a class that reads a line from the .csv file, parse the input string into its constituent values, compute an average of the scores for each student, convert the average into a letter grade, and print the student ID, scores, average of the scores and the letter equivalent of the average for each student. Also, track min and max averages, and calculate the class average. After all student data has been processed, print a summary for the class. Your output should match the formatting in the example output below in both the terminal and an output file named gradesout.txt**.

GRADE is defined as follows:

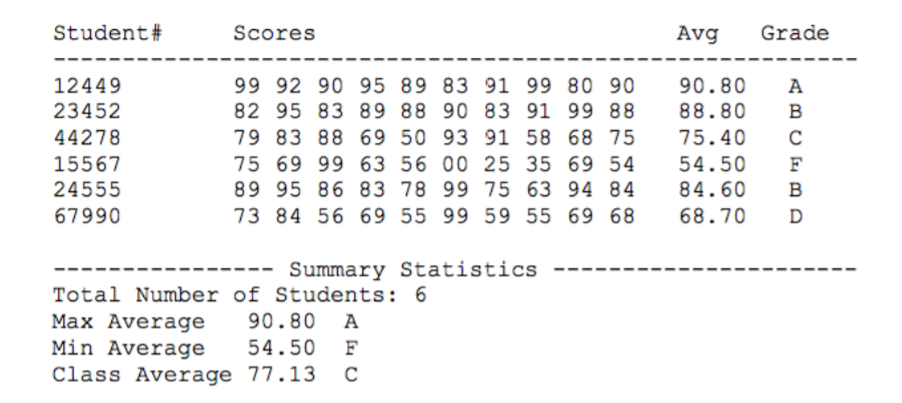
[100..90]: A

[90..80]: B

[80..70]: C

[70..60]: D

[60..0]: F

****

import java.util.Scanner;

import java.io.\*;

import java.text.DecimalFormat;

import java.util.Arrays;

public class Main{

public static String Grade(double avg){

if(avg>=90 && avg<100)

return "A";

else if(avg>=80 && avg<90)

return "B";

else if(avg>=70 && avg<80)

return "C";

else if(avg>=60 && avg<70)

return "D";

else

return "F";

}

public static void main(String []arggs){

DecimalFormat dec=new DecimalFormat(".00");

try{

File a= new File("score.csv");

Scanner sc= new Scanner(a);

FileWriter out= new FileWriter("gradesout.txt");

String og\_rows ; //og\_rows= Original complete row

int i=0;

String Separray[];//Each row seperated array

int Studentid[]= new int[6];

String Scores[]=new String[6];// for Score column\_output

double Average[]=new double[6];// for average column\_output

String Grade[]= new String[6];// for grade\_column

double avg\_sum=0;

while(sc.hasNext()){

double sum= 0;

og\_rows= sc.nextLine();

Separray= og\_rows.split(","); //reading csv row

//student# column

Studentid[i]= Integer.parseInt(Separray[0]);

//scores column

Scores[i]=""; //removing null in index 0

for(int j=1;j<Separray.length;j++){

Scores[i]+= " "+Separray[j];//String concatenation

sum+= Double.parseDouble(Separray[j]);

}

//Average column

Average[i]=sum/(Separray.length-1);

//Grade column

Grade[i]= Grade(Average [i]);

//System.out.println(Grade[i]);

//For class average

avg\_sum+= Average[i];

i++;

}

//for output

System.out.println("Student#\t\t\tScores\t\t\t\t\tAvg\t\tGrade\n"+

"--------------------------------------------------------------------------------------------------------");//displaying in terminal

out.write("Student#\t\t\tScores\t\t\t\t\tAvg\t\t Grade\n"+

"------------------------------------------------------------------------------\n");//displaying in file

//data dislay

for (int s =0;s<6;s++){

out.write(Studentid[s]+"\t\t\t "+Scores[s]+"\t\t\t"+dec.format(Average[s]) +"\t\t "+Grade[s]+"\n");//file

System.out.println(Studentid[s]+"\t\t\t "+Scores[s]+"\t\t\t"+dec.format(Average[s]) +"\t\t "+Grade[s]+"\n");//terminal

}

//for min max

Arrays.sort(Average);

//for terminal

System.out.println("--------------------Summary Stats----------------------------------------------------------------------");

System.out.println("Total number of students: "+Studentid.length);

System.out.println("Max Average: "+dec.format(Average[5])+"\t"+Grade(Average[5]));

System.out.println("Min Average: "+dec.format(Average[0])+"\t"+Grade(Average[0]));

System.out.println("Class Average: "+dec.format(avg\_sum/6)+"\t"+Grade(avg\_sum/6));

//for file

out.write("--------------------Summary Stats--------------------------------------------\n");

out.write("Total number of students: "+Studentid.length+"\n");

out.write("Max Average: "+dec.format(Average[5])+"\t"+Grade(Average[5])+"\n");

out.write("Min Average: "+dec.format(Average[0])+"\t"+Grade(Average[0])+"\n");

out.write("Class Average: "+dec.format(avg\_sum/6)+"\t"+Grade(avg\_sum/6)+"\n");

out.close();

sc.close();

}

catch(Exception e){ System.out.println(e.getMessage()); }

}

}

**In Terminal:**

