# Homework

**Grade 11 Review 7a – More File Output and Input**

1. A class has 10 students, and each student has to write 5 tests in a semester. A 2D array can be used to store the score of all tests for each student:



|  | 0 | 1 | 2 | 3 | 4 |
| --- | --- | --- | --- | --- | --- |
| 0 |  |  |  |  |  |
| 1 |  |  |  |  |  |
| 2 |  |  |  |  |  |
| 3 |  |  |  |  |  |
| 4 |  |  |  |  |  |
| 5 |  |  |  |  |  |
| 6 |  |  |  |  |  |
| 7 |  |  |  |  |  |
| 8 |  |  |  |  |  |
| 9 |  |  |  |  |  |

Write a program that does the following

* 1. Declare and create a 2D array to store the score of 5 tests of each of 10 students
  2. Read input from a text file “marks.txt” to fill in the array
  3. Calculate the average of the five tests for each student, and output the results to the file “result.txt”
  4. Find out the student with the highest overall average, and output the student number to the file “result.txt”
  5. Calculate the class average of each test and output the result to file “result.txt”

| /\*  \* Program name: ClassStudentMarks.java  \*  \* By: Lucas Chow (Last edited: 2022-10-03)  \*  \* ICS4U1 - 04\_Gr11Review  \*  \* This program StudentMarks.java reads a text file "marks.txt",  \* reading the classes' student count and test in the semester,  \* then creates a 2d array with those respective dimensions.  \* It calculates the average of the 5 test per student, finds the  \* student with highest overall average, and calculates the class  \* average of each test, outputting it to result.txt  \*  \*/  import java.io.\*;  public class ClassStudentMarks  {  public static void main(String[] args)  {  //initializing variables and objects  String fileNameInput = "marks.txt";  String fileNameOutput = "result.txt";  String[] nextLine;  int marksArrStudents;  int marksArrTest;  double avTotal;  double highestScore;  int highestScoreIndex;  try  {  //creating BufferedWriter  BufferedWriter out = new BufferedWriter(new FileWriter(fileNameOutput,false));  BufferedReader in = new BufferedReader(new FileReader(fileNameInput));  marksArrStudents = Integer.parseInt(in.readLine());  marksArrTest = Integer.parseInt(in.readLine());    //creating arrays to store the data for the marks, averages, and test respectively  double[][] marksArray = new double[marksArrStudents][marksArrTest];  double[] averageArray = new double[marksArrStudents];  double[] testAverage = new double[marksArrTest];    //looping through the array to read the text file and put it in the array  for (int x = 0; x < marksArrStudents; x++)  {  nextLine = in.readLine().split(" ");  for (int i = 0; i < marksArrTest; i++)  {  marksArray[x][i] = Double.parseDouble(nextLine[i]);  }  }      //looping through it again calculating the averages  for (int i = 0; i < marksArrStudents; i++)  {  avTotal = 0;  for (int a = 0; a < marksArrTest; a++)  {  avTotal += marksArray[i][a];  }  averageArray[i] = avTotal / marksArrTest;  }      //printing out the averages  for (int i = 0; i < marksArrStudents; i++)  {  out.write(String.format("Student %d has average %.2f\n",(i+1),averageArray[i]));  }    //finding the highestScore  highestScore = averageArray[0];  highestScoreIndex = 1;  for (int i = 0; i < marksArrStudents; i++)  {  if (averageArray[i] > highestScore)  {  highestScore = averageArray[i];  highestScoreIndex = (i+1);  }    }    //output writes the student with the highest average to result.txt  out.write(String.format("\nStudent %d has the highest average of %.2f\n\n",highestScoreIndex, highestScore));    //loops through to find test averages  for (int i = 0; i < marksArrTest; i++)  {  for (int a = 0; a < marksArrStudents; a++)  {  testAverage[i] += marksArray[a][i];  }  testAverage[i] = testAverage[i] / marksArrStudents;  }      //writes the test averages  for (int i = 0; i < marksArrTest; i++)  {  out.write(String.format("Test %d had a class average of %.2f\n",(i+1),testAverage[i]));  }    //closes the out and in objects from bufferedWriter and bufferedReader respectively  out.close();  in.close();    //If everything ran succesfully, prints confirmation message  System.out.println("Everything was printed out succesfully! Check out \"marks.txt\" for the original file input and \"result.txt\" for the results!");  }    //catching file error exception  catch (IOException e)  {  //printing out the error  System.out.println(e + " Problem reading " + fileNameInput);  }    }  } |
| --- |
| marks.txt:    result.txt: |

1. **PrintLines.java**: Write a program that read each line in a file “line.txt”, then output them to the standard output.

| /\*  \* Program name: PrintLines.java  \*  \* By: Lucas Chow (Last edited: 2022-10-03)  \*  \* ICS4U1 - 04\_Gr11Review  \*  \* This program takes in the file "line.txt" and outputs the content in standard output  \*  \*/  //importing BufferedReader  import java.io.\*;  public class PrintLines  {  //start of main method  public static void main(String[] args)  {  //initializing variables and objects  String fileName = "line.txt";  String lineIn;  try  {  //creating BufferedReader  BufferedReader fileReader = new BufferedReader(new FileReader(fileName));    //getting input of first line  lineIn = fileReader.readLine();    //while input is not blank (null in this case)  while (lineIn != null)  {  //printing out the lines, and getting another line  System.out.println(lineIn);  lineIn = fileReader.readLine();  }    //closing file  fileReader.close();  }  //execptiong for file error  catch (IOException e)  {  //printing out the error  System.out.println(fileName +" error " + e);  }  }  } |
| --- |
| Input: (line.txt)    Output: |

1. **PrintAllChar.java**: Write a program that reads all lines in the file “allChar.txt”, then prints each character on each line on separate line, excluding the spaces, ‘\n’ and ‘\r’

| /\*  \* Program name: PrintAllChars.java  \*  \* By: Lucas Chow (Last edited: 2022-10-03)  \*  \* ICS4U1 - 04\_Gr11Review  \*  \* This program takes in a file called "allChar.txt" and prints each character per line  \*  \*/  //importing BufferedReader  import java.io.\*;  public class PrintAllChar  {  public static void main(String[] args)  {  //initializing variables and objects  String fileName = "allChar.txt";  int nextLine;  try  {  //creating BufferedReader  BufferedReader in = new BufferedReader(new FileReader(fileName));  nextLine = in.read();    //while not blank line  while (nextLine != -1)  {  //checking if char is whitespace  if (!(Character.isWhitespace((char)nextLine)))  {  //if not whitespace, prints out char  System.out.println((char)nextLine);  }    //reads next line  nextLine = in.read();  }  //closes file  in.close();  }    //catches exception of file name  catch (IOException e)  {  //prints out the erro and filename  System.out.println(e + " problem reading "+fileName);  }    }  } |
| --- |
| input: (allChar.txt)    output: |

1. **ReverseLines.java:** In the text file “reverse.txt”, the first line contains an integer which indicates the number of lines that follow. You program should read each one of these lines and then output them in reverse order. (hint: use an array)

| /\*  \* Program name: ReverseLines.java  \*  \* By: Lucas Chow (Last edited: 2022-10-03)  \*  \* ICS4U1 - 04\_Gr11Review  \*  \* This program reads reverse.txt. The first line is a number  \* on how many lines of text, creating an array, and  \* getting the values of each line and printing it out in backward order  \* (First element to get it will be last element to get out)  \*  \*/  //importing BufferedReader  import java.io.\*;  public class ReverseLines  {  public static void main(String[] args)  {  //initializing variables and objects  String fileName = "reverse.txt";  int numOfLines;  try  {  //creating BufferedReader  BufferedReader in = new BufferedReader(new FileReader(fileName));    //getting the first input which is a number, setting it to be the length of arr linesOfText  numOfLines = Integer.parseInt(in.readLine());    //creating array of indexes numOfLines  String[] linesOfText = new String[numOfLines];    //loops through array, reading lines for each index  for (int i = 0; i < numOfLines; i++)  {  linesOfText[i] = in.readLine();  }    //loops through array backwards and prints out the elements inside  for (int i = (numOfLines-1); i >= 0; i--)  {  System.out.println(linesOfText[i]);  }    //closing bufferedReader  in.close();  }    //catches error execption  catch (IOException e)  {    //prints out the error  System.out.println(e + " Problem reading " + fileName);  }    }  } |
| --- |
| input:    output: |