

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Programme** | **:** | **B,Tech(CSE)** | **Semester** | **:** | **Winter 2020-21** |
| **Course** | **:** | **Java Programming** | **Code** | **:** | **CSE1007** |
| **Faculty** | **:** | **Rajarajeshwari S** | **Slot** | **:** | **A1** |
| **Name** | **:** | **Prakriti Sharma** | **Reg. No.** | **:** | **19BCE1655** |

**DA2**

**LINK TO THE VIDEO :**

**https://drive.google.com/file/d/1upOSSFPuwahqrvuN514A0uDbMxdlHAZs/view?usp=sharing**

<https://drive.google.com/file/d/1upOSSFPuwahqrvuN514A0uDbMxdlHAZs/view?usp=sharing>

**THE DOCUMENT HAS :**

**1.ANSWER TO THE CLASS DEFINITION PART.**

**2.SCREENSHOT OF OUTPUT.**

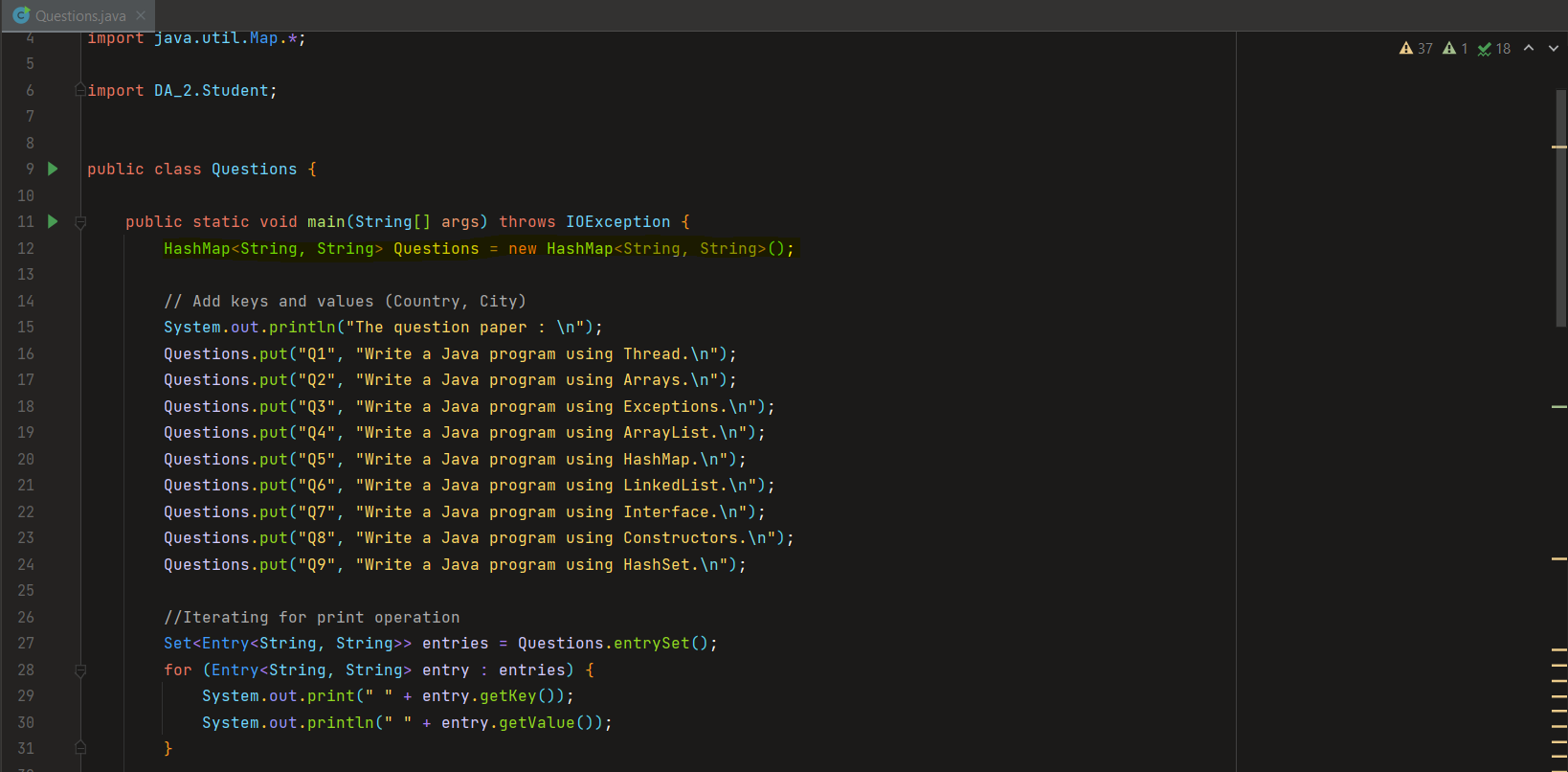
**3.PASTED CODE.**

**Link to the zip file : https://drive.google.com/file/d/1D1qAeyRilhIU2pgNF981Nv8KqXk8dTY6/view?usp=sharing**

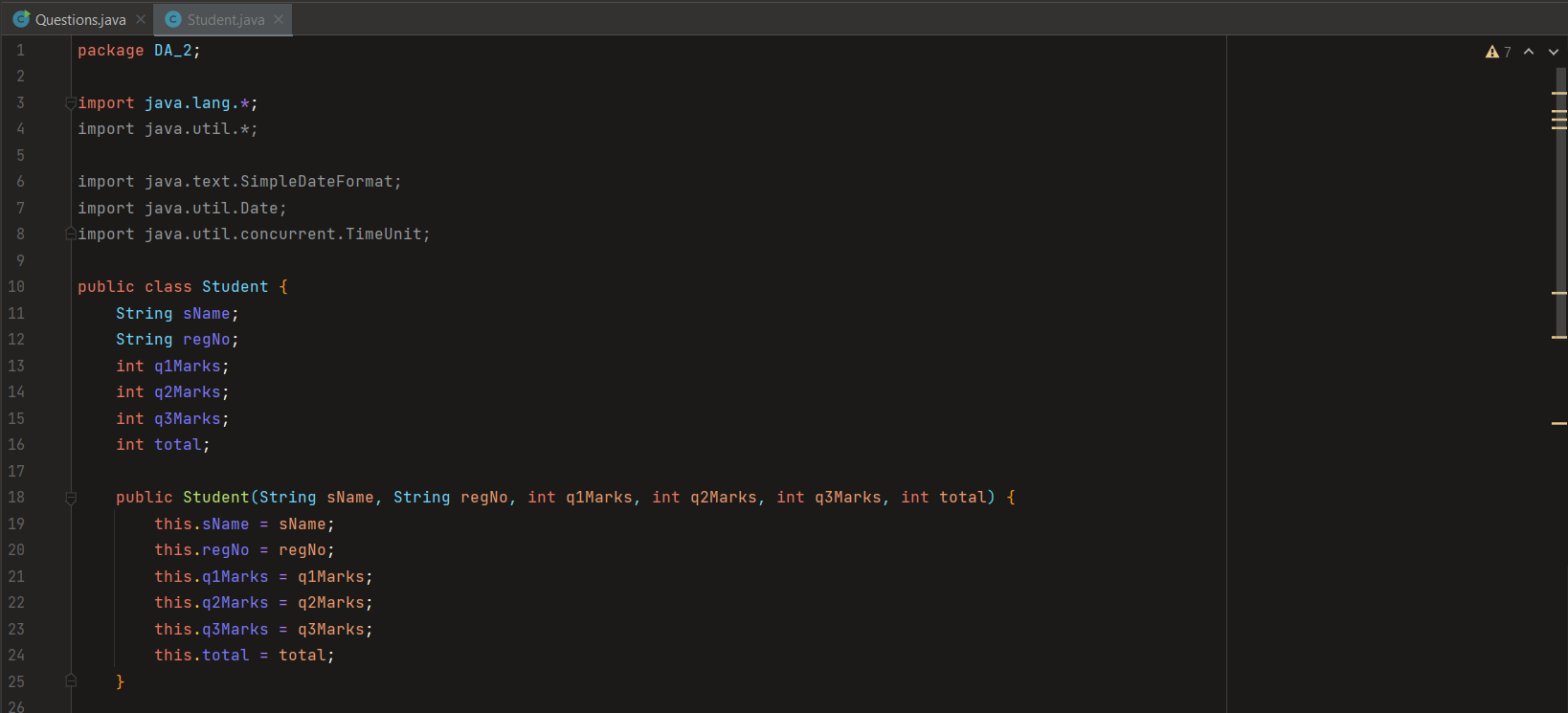
**1.ANSWER TO THE CLASS DEFINITION PART.**

Consider the online examination conducted by code tantra.

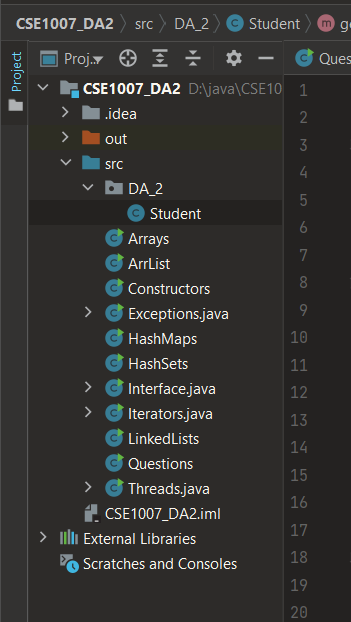
* Set of questions are maintained using HASH MAP/HASH TABLE(like Question Number ( Q1,Q2,Q3…Q9) as a key and Question text as a value). Question text will be like Java program using thread, java program using exception, java program using arraylist.



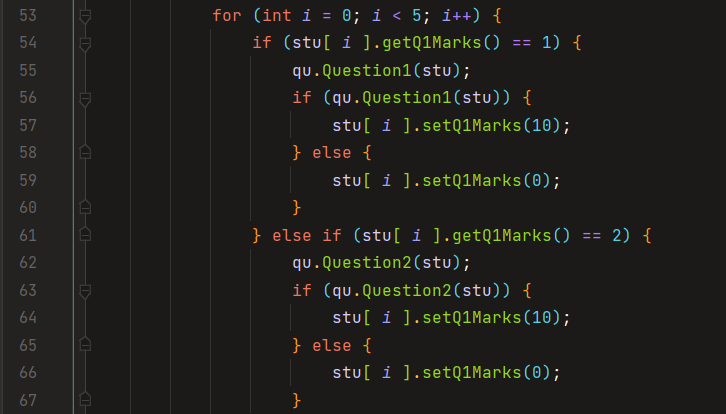
* Create Student class with the details like Reg no, and array of question numbers (maximum three questions of any choice.), marks array, total marks. ( question number should not be repeated for a single student)

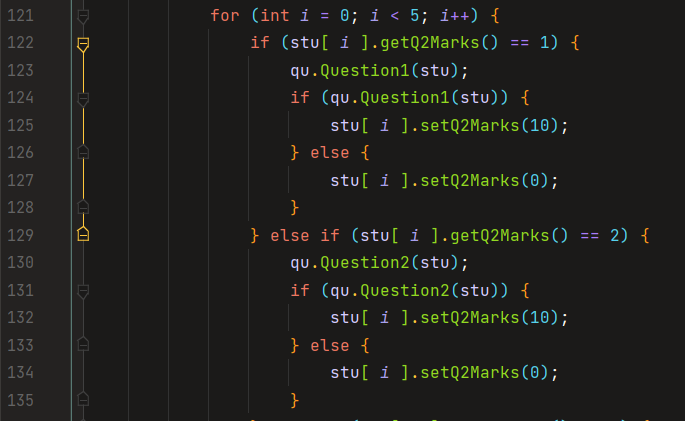


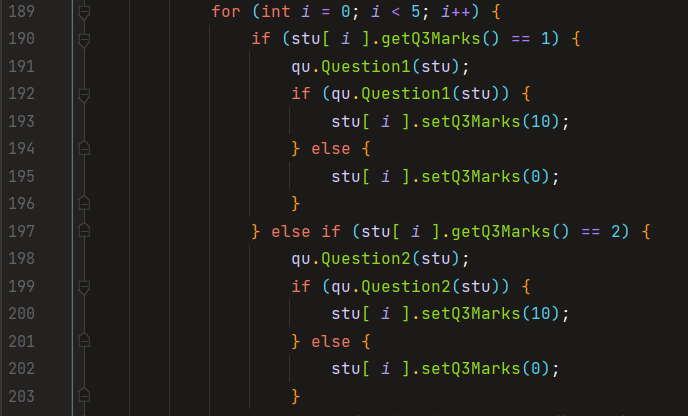
* Create a text file by every student which comprised of the Java code for all three question. first line Question number and next line onwards answer.



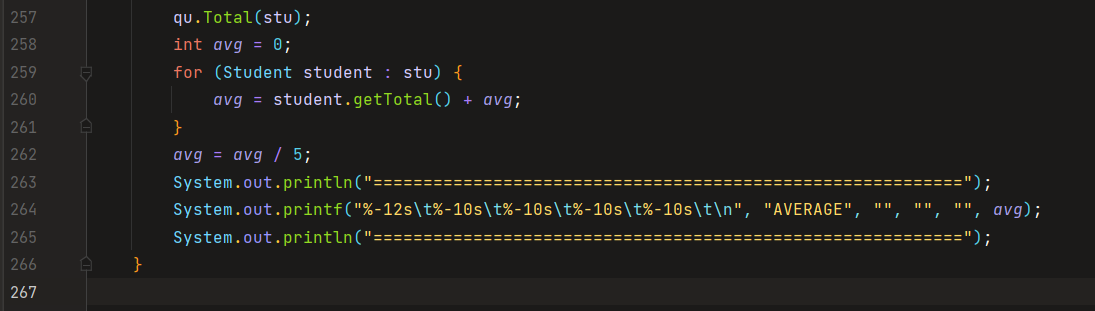
* Evaluate the answer script using the keywords in the question paper. For example, If the question comprised of java program using thread, search for the keyword thread in the answer. If it is there give the marks as 10. If the answer contains “public static void main(strings[]) and no matching keywords the n marks as 5. if the answer does not have either “public static void main( string a[]) and matching keywords , then marks as 0.



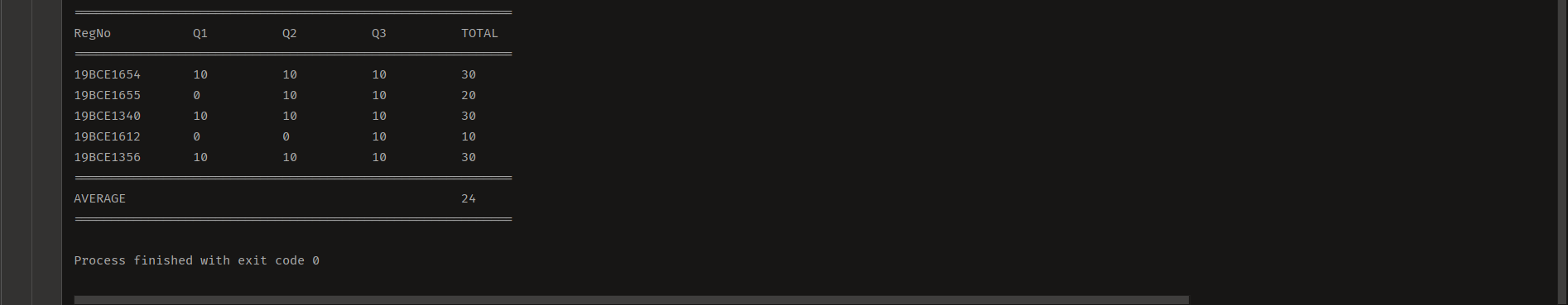
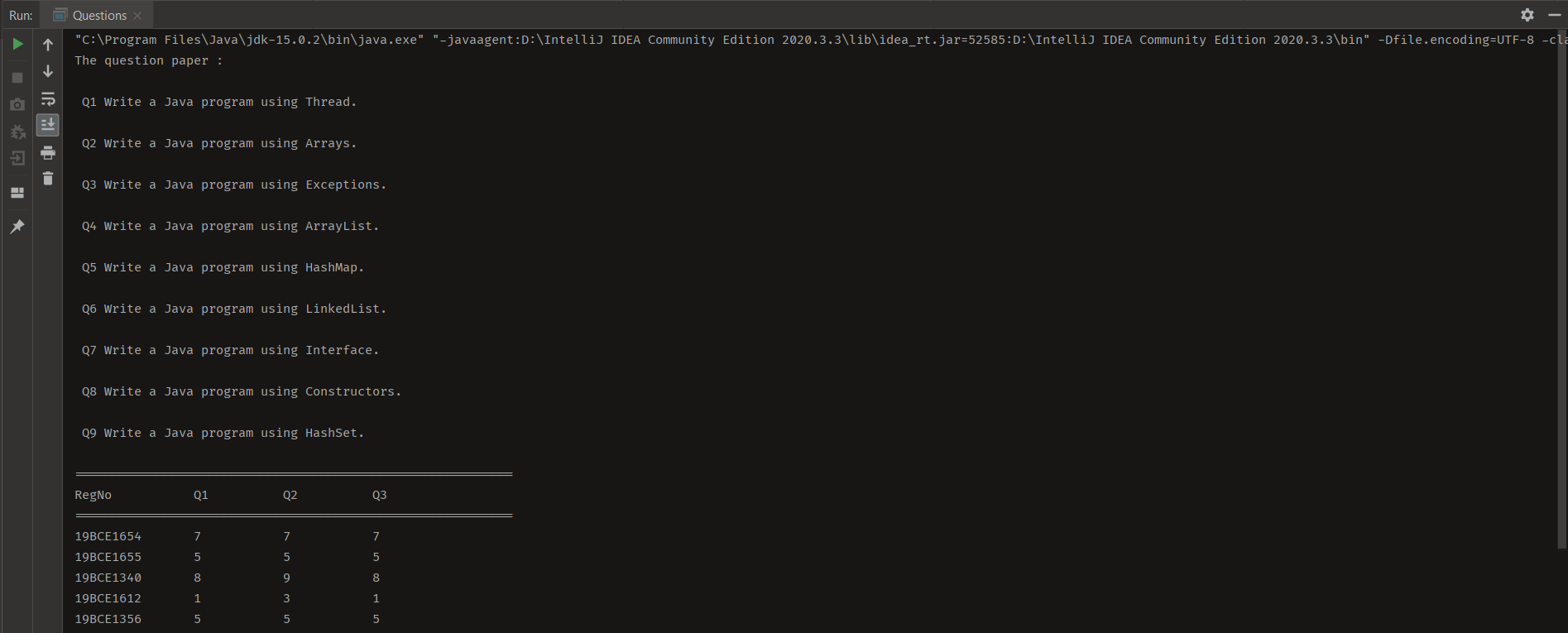




* Store the marks and total marks of every student in the student class. Print the mark list of all students and display the class average.



**3.SCREENSHOT OF OUTPUT.**



**4.PASTED CODE.**

CODE:

Questions.java

import java.io.\*;  
import java.util.HashMap;  
import java.util.\*;  
import java.util.Map.\*;  
  
import DA\_2.Student;  
  
  
public class Questions {  
  
 public static void main(String[] args) throws IOException {  
 HashMap<String, String> Questions = new HashMap<String, String>();  
  
 // Add keys and values (Country, City)  
 System.out.println("The question paper : \n");  
 Questions.put("Q1", "Write a Java program using Thread.\n");  
 Questions.put("Q2", "Write a Java program using Arrays.\n");  
 Questions.put("Q3", "Write a Java program using Exceptions.\n");  
 Questions.put("Q4", "Write a Java program using ArrayList.\n");  
 Questions.put("Q5", "Write a Java program using HashMap.\n");  
 Questions.put("Q6", "Write a Java program using LinkedList.\n");  
 Questions.put("Q7", "Write a Java program using Interface.\n");  
 Questions.put("Q8", "Write a Java program using Constructors.\n");  
 Questions.put("Q9", "Write a Java program using HashSet.\n");  
  
 //Iterating for print operation  
 Set<Entry<String, String>> entries = Questions.entrySet();  
 for (Entry<String, String> entry : entries) {  
 System.out.print(" " + entry.getKey());  
 System.out.println(" " + entry.getValue());  
 }  
  
 //Creating object  
 Questions qu = new Questions();  
  
 Student[] stu = new Student[ 5 ];  
 stu[ 0 ] = new Student("ABC", "19BCE1654", 0, 0, 0, 0);  
 stu[ 1 ] = new Student("DEF", "19BCE1655", 0, 0, 0, 0);  
 stu[ 2 ] = new Student("GHI", "19BCE1340", 0, 0, 0, 0);  
 stu[ 3 ] = new Student("JKL", "19BCE1612", 0, 0, 0, 0);  
 stu[ 4 ] = new Student("MNO", "19BCE1356", 0, 0, 0, 0);  
  
 Random rand = new Random();  
 System.out.println("===========================================================");  
 System.out.printf("%-12s\t%-10s\t%-10s\t%-10s\t\n", "RegNo", "Q1", "Q2", "Q3");  
 System.out.println("===========================================================");  
  
 for (int *j* = 0; *j* < stu.length; *j*++) {  
 int *n* = qu.generate(1,10);  
 stu[ *j* ].setQ1Marks(*n*);  
 *n* = qu.generate(1,10);  
 stu[ *j* ].setQ2Marks(*n*);  
 *n* = qu.generate(1,10);  
 stu[ *j* ].setQ3Marks(*n*);  
 System.out.printf("%-12s\t%-10s\t%-10s\t%-10s\t\n",stu[*j*].getRegNo(),  
 stu[*j*].getQ3Marks(),stu[*j*].getQ2Marks(),stu[*j*].getQ3Marks());  
  
 for (int *i* = 0; *i* < 5; *i*++) {  
 if (stu[ *i* ].getQ1Marks() == 1) {  
 qu.Question1(stu);  
 if (qu.Question1(stu)) {  
 stu[ *i* ].setQ1Marks(10);  
 } else {  
 stu[ *i* ].setQ1Marks(0);  
 }  
 } else if (stu[ *i* ].getQ1Marks() == 2) {  
 qu.Question2(stu);  
 if (qu.Question2(stu)) {  
 stu[ *i* ].setQ1Marks(10);  
 } else {  
 stu[ *i* ].setQ1Marks(0);  
 }  
 } else if (stu[ *i* ].getQ1Marks() == 3) {  
 qu.Question3(stu);  
 if (qu.Question3(stu)) {  
 stu[ *i* ].setQ1Marks(10);  
 } else {  
 stu[ *i* ].setQ1Marks(0);  
 }  
 } else if (stu[ *i* ].getQ1Marks() == 4) {  
 qu.Question4(stu);  
 if (qu.Question4(stu)) {  
 stu[ *i* ].setQ1Marks(10);  
 } else {  
 stu[ *i* ].setQ1Marks(0);  
 }  
 } else if (stu[ *i* ].getQ1Marks() == 5) {  
 qu.Question5(stu);  
 if (qu.Question5(stu)) {  
 stu[ *i* ].setQ1Marks(10);  
 } else {  
 stu[ *i* ].setQ1Marks(0);  
 }  
 } else if (stu[ *i* ].getQ1Marks() == 6) {  
 qu.Question6(stu);  
 if (qu.Question6(stu)) {  
 stu[ *i* ].setQ1Marks(10);  
 } else {  
 stu[ *i* ].setQ1Marks(0);  
 }  
 } else if (stu[ *i* ].getQ1Marks() == 7) {  
 qu.Question7(stu);  
 if (qu.Question7(stu)) {  
 stu[ *i* ].setQ1Marks(10);  
 } else {  
 stu[ *i* ].setQ1Marks(0);  
 }  
 } else if (stu[ *i* ].getQ1Marks() == 8) {  
 qu.Question8(stu);  
 if (qu.Question8(stu)) {  
 stu[ *i* ].setQ1Marks(10);  
 } else {  
 stu[ *i* ].setQ1Marks(0);  
 }  
 } else if (stu[ *i* ].getQ1Marks() == 9) {  
 qu.Question9(stu);  
 if (qu.Question9(stu)) {  
 stu[ *i* ].setQ1Marks(10);  
 } else {  
 stu[ *i* ].setQ1Marks(0);  
 }  
 }  
 }  
  
  
 for (int *i* = 0; *i* < 5; *i*++) {  
 if (stu[ *i* ].getQ2Marks() == 1) {  
 qu.Question1(stu);  
 if (qu.Question1(stu)) {  
 stu[ *i* ].setQ2Marks(10);  
 } else {  
 stu[ *i* ].setQ2Marks(0);  
 }  
 } else if (stu[ *i* ].getQ2Marks() == 2) {  
 qu.Question2(stu);  
 if (qu.Question2(stu)) {  
 stu[ *i* ].setQ2Marks(10);  
 } else {  
 stu[ *i* ].setQ2Marks(0);  
 }  
 } else if (stu[ *i* ].getQ2Marks() == 3) {  
 qu.Question3(stu);  
 if (qu.Question3(stu)) {  
 stu[ *i* ].setQ2Marks(10);  
 } else {  
 stu[ *i* ].setQ2Marks(0);  
 }  
 } else if (stu[ *i* ].getQ2Marks() == 4) {  
 qu.Question4(stu);  
 if (qu.Question4(stu)) {  
 stu[ *i* ].setQ2Marks(10);  
 } else {  
 stu[ *i* ].setQ2Marks(0);  
 }  
 } else if (stu[ *i* ].getQ2Marks() == 5) {  
 qu.Question5(stu);  
 if (qu.Question5(stu)) {  
 stu[ *i* ].setQ2Marks(10);  
 } else {  
 stu[ *i* ].setQ2Marks(0);  
 }  
 } else if (stu[ *i* ].getQ2Marks() == 6) {  
 qu.Question6(stu);  
 if (qu.Question6(stu)) {  
 stu[ *i* ].setQ2Marks(10);  
 } else {  
 stu[ *i* ].setQ2Marks(0);  
 }  
 } else if (stu[ *i* ].getQ2Marks() == 7) {  
 qu.Question7(stu);  
 if (qu.Question7(stu)) {  
 stu[ *i* ].setQ2Marks(10);  
 } else {  
 stu[ *i* ].setQ2Marks(0);  
 }  
 } else if (stu[ *i* ].getQ2Marks() == 8) {  
 qu.Question8(stu);  
 if (qu.Question8(stu)) {  
 stu[ *i* ].setQ2Marks(10);  
 } else {  
 stu[ *i* ].setQ2Marks(0);  
 }  
 } else if (stu[ *i* ].getQ2Marks() == 9) {  
 qu.Question9(stu);  
 if (qu.Question9(stu)) {  
 stu[ *i* ].setQ2Marks(10);  
 } else {  
 stu[ *i* ].setQ2Marks(0);  
 }  
 }  
 }  
  
  
 for (int *i* = 0; *i* < 5; *i*++) {  
 if (stu[ *i* ].getQ3Marks() == 1) {  
 qu.Question1(stu);  
 if (qu.Question1(stu)) {  
 stu[ *i* ].setQ3Marks(10);  
 } else {  
 stu[ *i* ].setQ3Marks(0);  
 }  
 } else if (stu[ *i* ].getQ3Marks() == 2) {  
 qu.Question2(stu);  
 if (qu.Question2(stu)) {  
 stu[ *i* ].setQ3Marks(10);  
 } else {  
 stu[ *i* ].setQ3Marks(0);  
 }  
 } else if (stu[ *i* ].getQ3Marks() == 3) {  
 qu.Question3(stu);  
 if (qu.Question3(stu)) {  
 stu[ *i* ].setQ3Marks(10);  
 } else {  
 stu[ *i* ].setQ3Marks(0);  
 }  
 } else if (stu[ *i* ].getQ3Marks() == 4) {  
 qu.Question4(stu);  
 if (qu.Question4(stu)) {  
 stu[ *i* ].setQ3Marks(10);  
 } else {  
 stu[ *i* ].setQ3Marks(0);  
 }  
 } else if (stu[ *i* ].getQ3Marks() == 5) {  
 qu.Question5(stu);  
 if (qu.Question5(stu)) {  
 stu[ *i* ].setQ3Marks(10);  
 } else {  
 stu[ *i* ].setQ3Marks(0);  
 }  
 } else if (stu[ *i* ].getQ3Marks() == 6) {  
 qu.Question6(stu);  
 if (qu.Question6(stu)) {  
 stu[ *i* ].setQ3Marks(10);  
 } else {  
 stu[ *i* ].setQ3Marks(0);  
 }  
 } else if (stu[ *i* ].getQ3Marks() == 7) {  
 qu.Question7(stu);  
 if (qu.Question7(stu)) {  
 stu[ *i* ].setQ3Marks(10);  
 } else {  
 stu[ *i* ].setQ3Marks(0);  
 }  
 } else if (stu[ *i* ].getQ3Marks() == 8) {  
 qu.Question8(stu);  
 if (qu.Question8(stu)) {  
 stu[ *i* ].setQ3Marks(10);  
 } else {  
 stu[ *i* ].setQ3Marks(0);  
 }  
 } else if (stu[ *i* ].getQ3Marks() == 9) {  
 qu.Question9(stu);  
 if (qu.Question9(stu)) {  
 stu[ *i* ].setQ3Marks(10);  
 } else {  
 stu[ *i* ].setQ3Marks(0);  
 }  
 }  
 }  
  
 }  
 qu.Total(stu);  
 int *avg* = 0;  
 for (Student student : stu) {  
 *avg* = student.getTotal() + *avg*;  
 }  
 *avg* = *avg* / 5;  
 System.out.println("===========================================================");  
 System.out.printf("%-12s\t%-10s\t%-10s\t%-10s\t%-10s\t\n", "AVERAGE", "", "", "", *avg*);  
 System.out.println("===========================================================");  
 }  
  
 public boolean Question1(Student[] stu) throws IOException {  
 File f1 = new File("D:\\java\\CSE1007\_DA2\\src\\Threads.java"); //Creation of File Descriptor for input file  
 String[] *words* = null; //Intialize the word Array  
 FileReader fr = new FileReader(f1); //Creation of File Reader object  
 BufferedReader br = new BufferedReader(fr); //Creation of BufferedReader object  
 String *s*;  
 String input = "thread"; // Input word to be searched  
 int *count* = 0; //Intialize the word to zero  
 while ((*s* = br.readLine()) != null) //Reading Content from the file  
 {  
 *words* = *s*.split(" "); //Split the word using space  
 for (String word : *words*) {  
 if (word.equals(input)) //Search for the given word  
 {  
 *count*++; //If Present increase the count by one  
 }  
 }  
 }  
 fr.close();  
 if(*count* != 0) //Check for count not equal to zero  
 {  
 return true;  
  
 } else {  
 return false;  
 }  
 }  
  
 public boolean Question2(Student[] stu) throws IOException {  
 File f1 = new File("D:\\java\\CSE1007\_DA2\\src\\Arrays.java"); //Creation of File Descriptor for input file  
 String[] *words* = null; //Intialize the word Array  
 FileReader fr = new FileReader(f1); //Creation of File Reader object  
 BufferedReader br = new BufferedReader(fr); //Creation of BufferedReader object  
 String *s*;  
 String input = "array"; // Input word to be searched  
 int *count* = 0; //Intialize the word to zero  
 while ((*s* = br.readLine()) != null) //Reading Content from the file  
 {  
 *words* = *s*.split(" "); //Split the word using space  
 for (String word : *words*) {  
 if (word.equals(input)) //Search for the given word  
 {  
 *count*++; //If Present increase the count by one  
 }  
 }  
 }  
 fr.close();  
 if(*count* != 0) //Check for count not equal to zero  
 {  
 return true;  
  
 } else {  
 return false;  
 }  
 }  
  
 public boolean Question3(Student[] stu) throws IOException {  
 File f1 = new File("D:\\java\\CSE1007\_DA2\\src\\Exceptions.java"); //Creation of File Descriptor for input file  
 String[] *words* = null; //Intialize the word Array  
 FileReader fr = new FileReader(f1); //Creation of File Reader object  
 BufferedReader br = new BufferedReader(fr); //Creation of BufferedReader object  
 String *s*;  
 String input = "exception"; // Input word to be searched  
 int *count* = 0; //Intialize the word to zero  
 while ((*s* = br.readLine()) != null) //Reading Content from the file  
 {  
 *words* = *s*.split(" "); //Split the word using space  
 for (String word : *words*) {  
 if (word.equals(input)) //Search for the given word  
 {  
 *count*++; //If Present increase the count by one  
 }  
 }  
 }  
 fr.close();  
 if(*count* != 0) //Check for count not equal to zero  
 {  
 return true;  
  
 } else {  
 return false;  
 }  
 }  
  
 public boolean Question4(Student[] stu) throws IOException {  
 File f1 = new File("D:\\java\\CSE1007\_DA2\\src\\ArrList.java"); //Creation of File Descriptor for input file  
 String[] *words* = null; //Intialize the word Array  
 FileReader fr = new FileReader(f1); //Creation of File Reader object  
 BufferedReader br = new BufferedReader(fr); //Creation of BufferedReader object  
 String *s*;  
 String input = "ArrayList"; // Input word to be searched  
 int *count* = 0; //Intialize the word to zero  
 while ((*s* = br.readLine()) != null) //Reading Content from the file  
 {  
 *words* = *s*.split(" "); //Split the word using space  
 for (String word : *words*) {  
 if (word.equals(input)) //Search for the given word  
 {  
 *count*++; //If Present increase the count by one  
 }  
 }  
 }  
 fr.close();  
 if(*count* != 0) //Check for count not equal to zero  
 {  
 return true;  
  
 } else {  
 return false;  
 }  
 }  
  
 public boolean Question5(Student[] stu) throws IOException {  
 File f1 = new File("D:\\java\\CSE1007\_DA2\\src\\HashMaps.java"); //Creation of File Descriptor for input file  
 String[] *words* = null; //Intialize the word Array  
 FileReader fr = new FileReader(f1); //Creation of File Reader object  
 BufferedReader br = new BufferedReader(fr); //Creation of BufferedReader object  
 String *s*;  
 String input = "HashMap"; // Input word to be searched  
 int *count* = 0; //Intialize the word to zero  
 while ((*s* = br.readLine()) != null) //Reading Content from the file  
 {  
 *words* = *s*.split(" "); //Split the word using space  
 for (String word : *words*) {  
 if (word.equals(input)) //Search for the given word  
 {  
 *count*++; //If Present increase the count by one  
 }  
 }  
 }  
 fr.close();  
 if(*count* != 0) //Check for count not equal to zero  
 {  
 return true;  
  
 } else {  
 return false;  
 }  
 }  
  
 public boolean Question6(Student[] stu) throws IOException {  
 File f1 = new File("D:\\java\\CSE1007\_DA2\\src\\LinkedLists.java"); //Creation of File Descriptor for input file  
 String[] *words* = null; //Intialize the word Array  
 FileReader fr = new FileReader(f1); //Creation of File Reader object  
 BufferedReader br = new BufferedReader(fr); //Creation of BufferedReader object  
 String *s*;  
 String input = "LinkedList"; // Input word to be searched  
 int *count* = 0; //Intialize the word to zero  
 while ((*s* = br.readLine()) != null) //Reading Content from the file  
 {  
 *words* = *s*.split(" "); //Split the word using space  
 for (String word : *words*) {  
 if (word.equals(input)) //Search for the given word  
 {  
 *count*++; //If Present increase the count by one  
 }  
 }  
 }  
 fr.close();  
 if(*count* != 0) //Check for count not equal to zero  
 {  
 return true;  
  
 } else {  
 return false;  
 }  
 }  
  
 public boolean Question7(Student[] stu) throws IOException {  
 File f1 = new File("D:\\java\\CSE1007\_DA2\\src\\Interface.java"); //Creation of File Descriptor for input file  
 String[] *words* = null; //Intialize the word Array  
 FileReader fr = new FileReader(f1); //Creation of File Reader object  
 BufferedReader br = new BufferedReader(fr); //Creation of BufferedReader object  
 String *s*;  
 String input = "interface"; // Input word to be searched  
 int *count* = 0; //Intialize the word to zero  
 while ((*s* = br.readLine()) != null) //Reading Content from the file  
 {  
 *words* = *s*.split(" "); //Split the word using space  
 for (String word : *words*) {  
 if (word.equals(input)) //Search for the given word  
 {  
 *count*++; //If Present increase the count by one  
 }  
 }  
 }  
 fr.close();  
 if(*count* != 0) //Check for count not equal to zero  
 {  
 return true;  
  
 } else {  
 return false;  
 }  
 }  
  
 public boolean Question8(Student[] stu) throws IOException {  
 File f1 = new File("D:\\java\\CSE1007\_DA2\\src\\Constructors.java"); //Creation of File Descriptor for input file  
 String[] *words* = null; //Intialize the word Array  
 FileReader fr = new FileReader(f1); //Creation of File Reader object  
 BufferedReader br = new BufferedReader(fr); //Creation of BufferedReader object  
 String *s*;  
 String input = "class"; // Input word to be searched  
 int *count* = 0; //Intialize the word to zero  
 while ((*s* = br.readLine()) != null) //Reading Content from the file  
 {  
 *words* = *s*.split(" "); //Split the word using space  
 for (String word : *words*) {  
 if (word.equals(input)) //Search for the given word  
 {  
 *count*++; //If Present increase the count by one  
 }  
 }  
 }  
 fr.close();  
 if(*count* != 0) //Check for count not equal to zero  
 {  
 return true;  
  
 } else {  
 return false;  
 }  
 }  
  
 public boolean Question9(Student[] stu) throws IOException {  
 File f1 = new File("D:\\java\\CSE1007\_DA2\\src\\HashSets.java"); //Creation of File Descriptor for input file  
 String[] *words* = null; //Intialize the word Array  
 FileReader fr = new FileReader(f1); //Creation of File Reader object  
 BufferedReader br = new BufferedReader(fr); //Creation of BufferedReader object  
 String *s*;  
 String input = "HashSet"; // Input word to be searched  
 int *count* = 0; //Intialize the word to zero  
 while ((*s* = br.readLine()) != null) //Reading Content from the file  
 {  
 *words* = *s*.split(" "); //Split the word using space  
 for (String word : *words*) {  
 if (word.equals(input)) //Search for the given word  
 {  
 *count*++; //If Present increase the count by one  
 }  
 }  
 }  
 fr.close();  
 if(*count* != 0) //Check for count not equal to zero  
 {  
 return true;  
  
 } else {  
 return false;  
 }  
 }  
  
 public void Total(Student[] stu){  
 System.out.println("===========================================================");  
 System.out.printf("%-12s\t%-10s\t%-10s\t%-10s\t%-10s\t\n", "RegNo", "Q1", "Q2", "Q3", "TOTAL");  
 System.out.println("===========================================================");  
 int *total*=0;  
 int m1,m2,m3;  
 m1=m2=m3=0;  
 for (int *i*=0;*i*<stu.length;*i*++) {  
 *total* = stu[*i*].getQ1Marks()+stu[*i*].getQ2Marks()+stu[*i*].getQ3Marks();  
 stu[*i*].setTotal(*total*);  
 System.out.printf("%-12s\t%-10s\t%-10s\t%-10s\t%-10s\t\n",stu[*i*].getRegNo(),  
 stu[*i*].getQ1Marks(),stu[*i*].getQ2Marks(),stu[*i*].getQ3Marks(),stu[*i*].getTotal());  
 }  
 }  
 public int generate(int min, int max) {  
 return (int) ((Math.random() \* (max - min)) + min);  
 }  
}

Student.java

package DA\_2;  
  
import java.lang.\*;  
import java.util.\*;  
  
import java.text.SimpleDateFormat;  
import java.util.Date;  
import java.util.concurrent.TimeUnit;  
  
public class Student {  
 String sName;  
 String regNo;  
 int q1Marks;  
 int q2Marks;  
 int q3Marks;  
 int total;  
  
 public Student(String sName, String regNo, int q1Marks, int q2Marks, int q3Marks, int total) {  
 this.sName = sName;  
 this.regNo = regNo;  
 this.q1Marks = q1Marks;  
 this.q2Marks = q2Marks;  
 this.q3Marks = q3Marks;  
 this.total = total;  
 }  
  
 public String getsName() {  
  
 return sName;  
 }  
  
 public void setsName(String sName) {  
  
 this.sName = sName;  
 }  
  
 public String getRegNo() {  
  
 return regNo;  
 }  
  
 public void setRegNo(String regNo) {  
  
 this.regNo = regNo;  
 }  
  
 public int getQ1Marks() {  
  
 return q1Marks;  
 }  
  
 public void setQ1Marks(int q1Marks) {  
  
 this.q1Marks = q1Marks;  
 }  
  
 public int getQ2Marks() {  
  
 return q2Marks;  
 }  
  
 public void setQ2Marks(int q2Marks) {  
  
 this.q2Marks = q2Marks;  
 }  
  
 public int getQ3Marks() {  
  
 return q3Marks;  
 }  
  
 public void setQ3Marks(int q3Marks) {  
  
 this.q3Marks = q3Marks;  
 }  
  
 public int getTotal() {  
  
 return total;  
 }  
  
 public void setTotal(int total) {  
  
 this.total = total;  
 }  
}