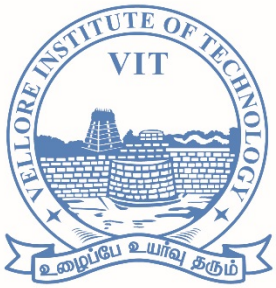


VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

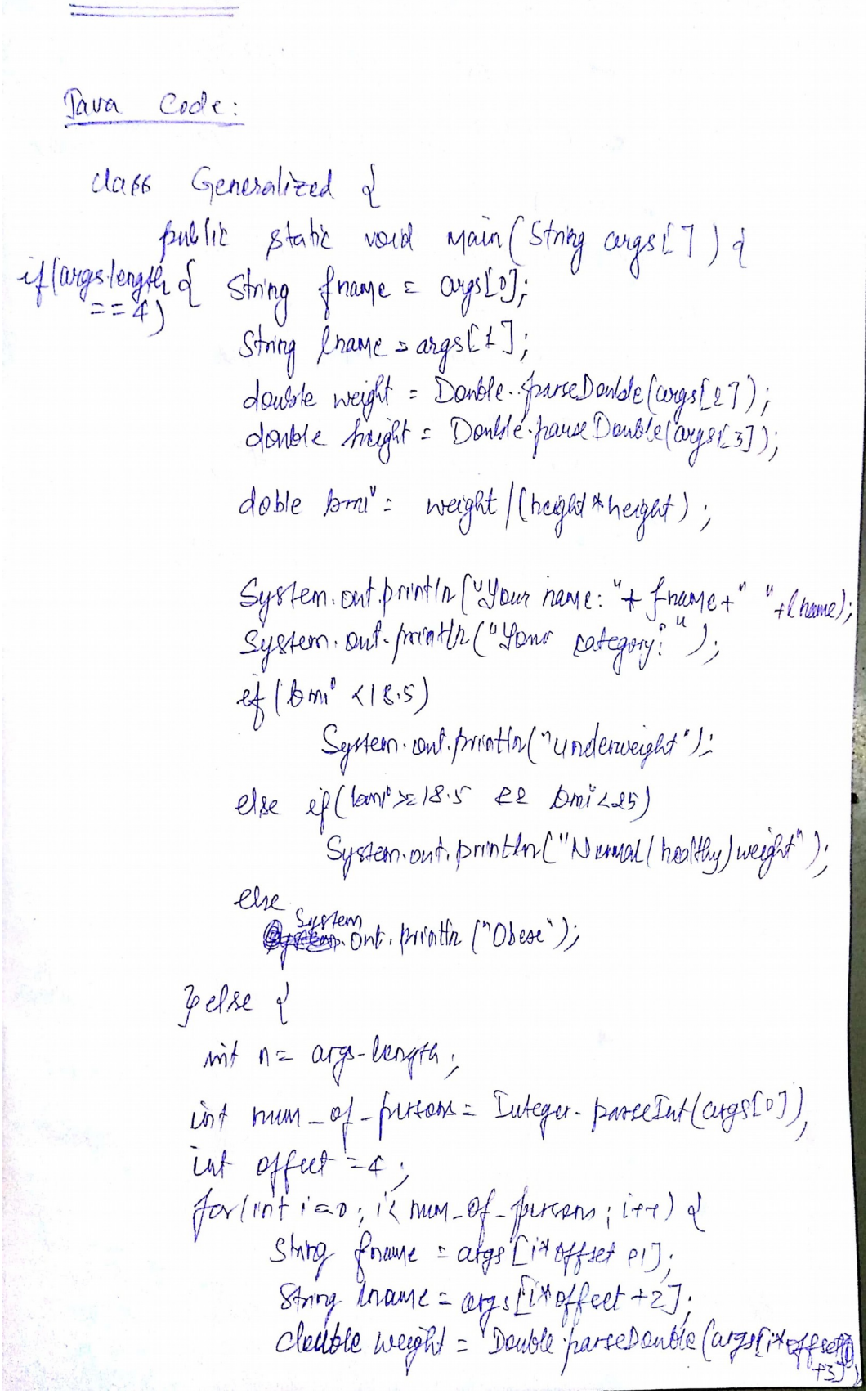
CHALLENGING LAB EXPERIMENTS- 1 to 5

Course- Advanced JAVA Programming
Slot- L25+L26
Faculty- Prof. Priya V

Submitted By:
Name- Krishna Kumar Mahto
Registration number- 16BIT0453

Question 1:

Generalized code for all Low + Medium + High levels:



Java Code:

```
class Generalized {
    public static void main(String args[]) {
if(args.length == 4) { String fname = args[0];
        String lname = args[1];
        double weight = Double.parseDouble(args[2]);
        double height = Double.parseDouble(args[3]);

        doble bmi = weight/(height*height);

        System.out.println("Your name: "+fname+" "+lname);
        System.out.println("Your category: ");
        if(bmi<18.5)
            System.out.println("Underweight");
        else if(bmi>=18.5 && bmi<25)
            System.out.println("Normal(healthy) weight");
        else
            System.out.println("Obese");
    }else {
        int n = args.length;
        int num_of_persons = Integer.parseInt(args[0]);
        int offset = 4;
        for(int i=0; i<num_of_persons; i++) {
            String fname = args[i*offset+1];
            String lname = args[i*offset+2];
            cloudble weight = Double.parseDouble(args[i*offset+3]);
```

```
double height = Double.parseDouble(args[1]);

double bmi = weight/(height * height);
System.out.println("Your name: "+fname+" "+lname);
if(bmi<18.5)
    System.out.println("underweight");
else if(bmi>=18.5 && bmi<25)
    System.out.println("Normal weight");
else if(bmi>=25 && bmi<30)
    System.out.println("Overweight");
else
    System.out.println("Obese class");
}
}
}
}
```

Outputs:

```
krish-thorcode@kkm-ubuntu:~/My_files/College_related_stuffs/Fall_2018-19/Courses
/Advanced JAVA/Lab/challenging_exercises/question1$ java Generalized krishna kum
ar 66 1.72
Your name: krishna kumar
Your Category: Normal(healthy weight)
```

```
krish-thorcode@kkm-ubuntu:~/My_files/College_related_stuffs/Fall_2018-19/Courses
/Advanced JAVA/Lab/challenging_exercises/question1$ java Generalized 4 krishna k
umar 66 1.72 ram manohar 89 1.5 some thing 15 1.78 anything nothing 78 1.73
Your name: krishna kumar
Your Category: Normal(healthy weight)
Your name: ram manohar
Your Category: Obese class
Your name: some thing
Your Category: Underweight
Your name: anything nothing
Your Category: Overweight
```

Question 2:

Generalized code for all Low + Medium + High levels:

Question - 2

Java code:

```
import java.util.Scanner;

class Generalized {
    public static void main(String args[]) {
        int n;
        Scanner scanner = new Scanner(System.in);
        n = Scanner.nextInt();

        int numBatches = 4;
        int numStudents = 70;
        int batchSize, remainingStudents, numGroupsOfFour;
        int[] coursesGroupOfFourOnly = new int[4*n];
        int[] batchesGroupOfFourOnly = new int[4*n];
        int count = 0;
        int[][][] studentMentors = new int[n][numBatches]
                                          [numStudents];

        for(int i=0; i<n; i++) {
            System.out.println("Course" + i + ":");
            for(int j = 0; j<numBatches; j++) {
                System.out.println("Enter num of class learners for
                    batch" + j + ":");
                batchSize = scanner.nextInt();

                numGroupsOfFour = batchSize/4;
                remainingStudents = batchSize - numGroupsOfFour*4;
```

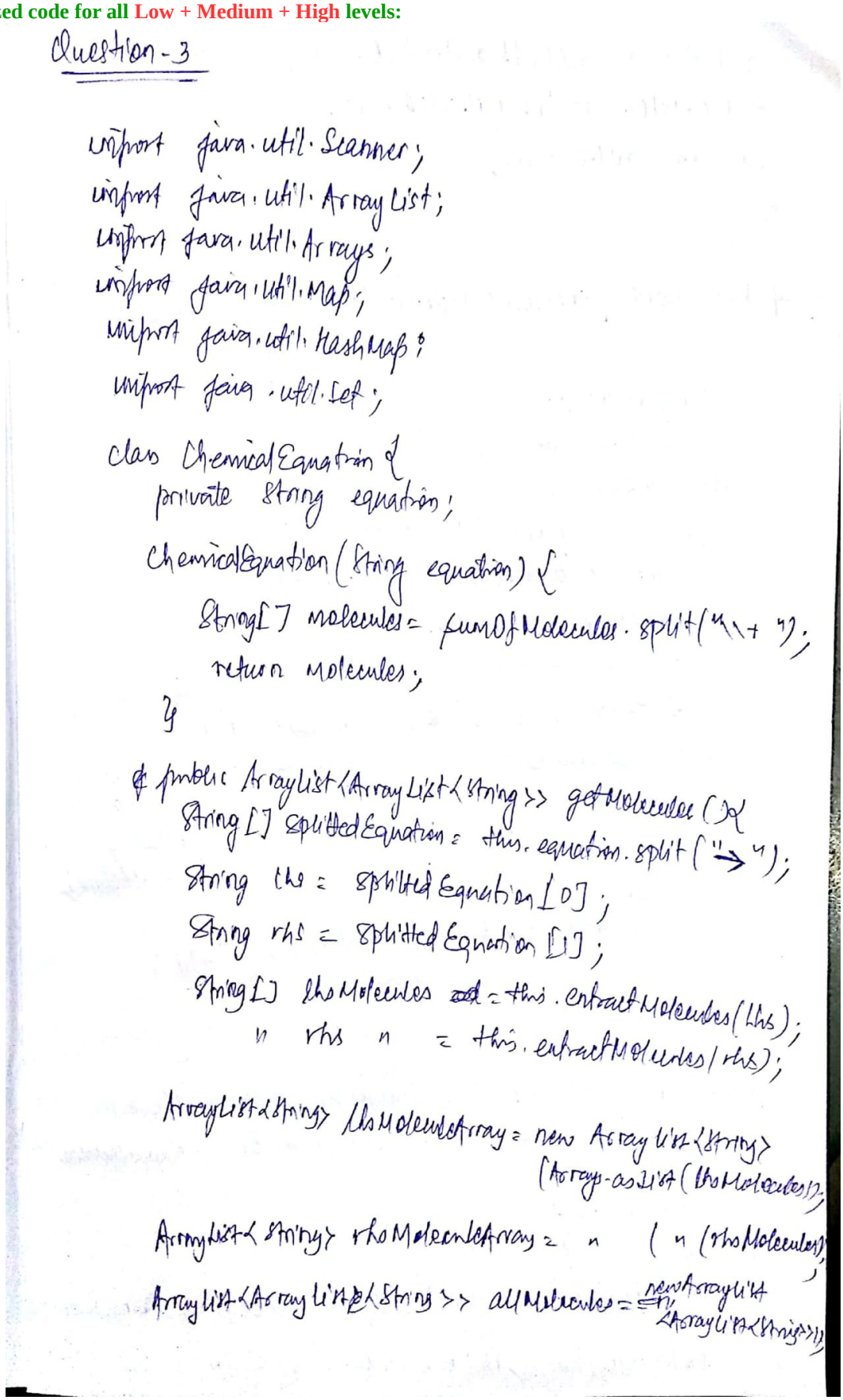
```
int k;
for (k = 0; k < numGroupsOfFour; k++)
    StudentMentors[i][j][k] = 4;
if (remainingStudents != 0)
    StudentMentors[i][j][k] = remainingStudents;
else {
    CoursesGroupOfFourOnly[count] = i;
    BatchesGroupOfFourOnly[count++] = j;
}
}
}
int iterator = 0;
if (BatchesGroupOfFourOnly.length != 0) {
    System.out.println("Batches with all groups of exactly
                        4");
    for (int course : courses GroupOfFourOnly) {
        if (iterator == count)
            break;
        System.out.println("Batches
        System.out.println("Course: " + course + ": ");
        System.out.println("Batch." + BatchesGroupOf
                            FourOnly[iterator++]);
    }
}
}
}
```

Outputs:

```
krish-thorcode@kkm-ubuntu:~/My_files/College_related_stuffs/Fall_2018-19/Courses
/Advanced JAVA/Lab/challenging_exercises/question2$ java Generalized
Enter the number of courses: 3
Course 0:
Enter number of slow learnings for batch 0: 23
Enter number of slow learnings for batch 1: 21
Enter number of slow learnings for batch 2: 15
Enter number of slow learnings for batch 3: 9
Course 1:
Enter number of slow learnings for batch 0: 24
Enter number of slow learnings for batch 1: 32
Enter number of slow learnings for batch 2: 11
Enter number of slow learnings for batch 3: 2
Course 2:
Enter number of slow learnings for batch 0: 22
Enter number of slow learnings for batch 1: 11
Enter number of slow learnings for batch 2: 8
Enter number of slow learnings for batch 3: 3
Batches with all groups of exactly 4:
Course 1: Batch 0
Course 1: Batch 1
Course 2: Batch 2
```

Question 3:

Generalized code for all Low + Medium + High levels:



Question - 3

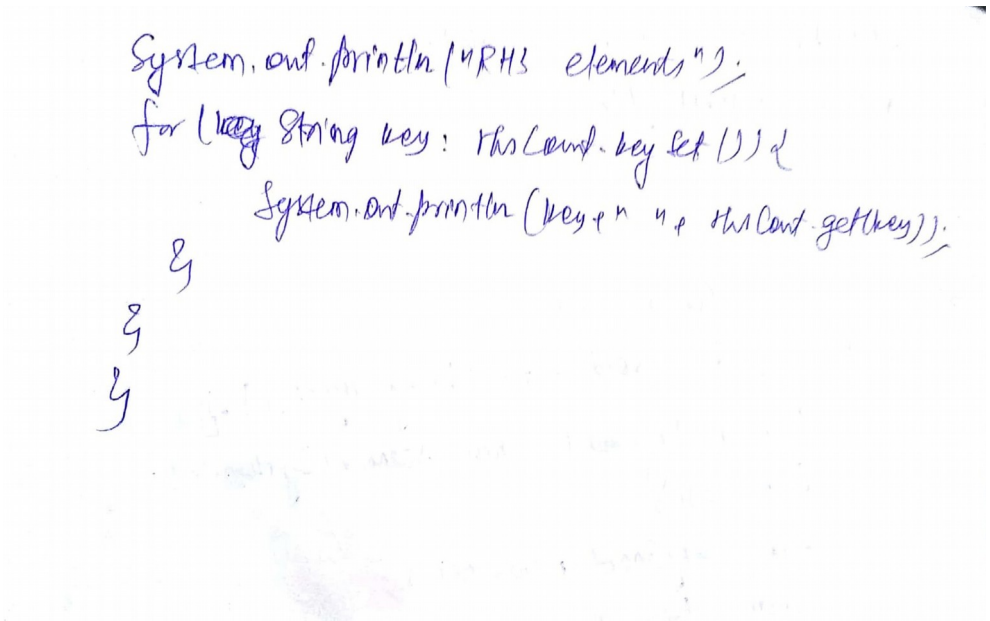
```
import java.util.Scanner;
import java.util.ArrayList;
import java.util.Arrays;
import java.util.Map;
import java.util.HashMap;
import java.util.Set;

class ChemicalEquation {
    private String equation;

    ChemicalEquation(String equation) {
        String[] molecules = sumOfMolecules.split("\\+");
        return molecules;
    }

    public ArrayList<ArrayList<String>> getMolecules() {
        String[] splittedEquation = this.equation.split("->");
        String lhs = splittedEquation[0];
        String rhs = splittedEquation[1];
        String[] lhsMolecules = this.extractMolecules(lhs);
        " rhs " = this.extractMolecules(rhs);

        ArrayList<String> lhsMoleculeArray = new ArrayList<String>
            (Arrays.asList(lhsMolecules));

        ArrayList<String> rhsMoleculeArray = " (" (rhsMolecules)
        ArrayList<ArrayList<String>> allMolecules = new ArrayList
            <ArrayList<String>>();
```

```
allMolecules.add(lhsMoleculeArray);
allMolecules.add(rhsMoleculeArray);
return allMolecules;
}

public void printNumOfMolecules(ArrayList<String> molecules,
                                String side){

    String molecule;
    char firstChar;
    int isDigit;
    System.out.println("Num of molecules "+side+": ");
    for(int i=0; i<molecules.size(); i++){
        molecule= molecules.get(i);
        firstChar = molecule.charAt(0);
        if(firstChar>='0' && firstChar<='9')
            System.out.println(molecule.substring(1,molecule.
                length())
                +"_"+
                +firstChar);
        else
            System.out.println("molecule"+"_"+1);
    }
}

public ArrayList<Map<String,Integer>> getElementCount(
    ArrayList<String> lhsMolecules, ArrayList<String> rhsMolecules)

{   String molecule;
    String element="";
    Map<String,Integer> lhsElements= new HashMap<String,Integer>
                                        ();
    Map<String,Integer> rhsElements= "  "  ";
```

```
char firstChar, ch;
int isDigit;

ArrayList<Map<String,Integer>> returnMapList = new ArrayList<Map
                                                <String,Integer>>();

~~for(int i=0; i<lhsMolecules.get(~~
for(int i=0; i<lhsMolecules.size(); i++){
    molecule = lhsMolecules.get(i);
    ~~System.out.println(molecule+"~~
    firstChar = molecule.charAt(0);
    if(firstChar>='0' && firstChar<='9')
        molecule = molecule.substring(1, molecule.length());
    ~~System.out.println(molecule+"~~

    for(int j=0; j<molecule.length();j++){
    if(molecule.charAt(j)>='0' && molecule.charAt(j)<='9'){
        lhsElements.put(element, lhsElements.get(element)+Character.
                        getNumericValue(molecule.charAt(j)));
        continue;
    }
    if(Character.isLowerCase(molecule.charAt(j))){
        element = molecule.substring(0-1,j+1);
        if(lhsElements.containsKey(element))
            lhsElements.put(element, lhsElements.get(element)
                            +Character.getNumericValue(firstChar));
        else
            lhsElements.put(element, lhsElements.get(element)-1
                            Character.getNumericValue(firstChar));
    }
```

```
else {
    element = molecule.substring(j, j+1);
    if (lhsElements.containsKey(element))
        lhsElements.put(element, lhsElements.get(element) +
                        Character.getNumericValue(firstChar));
    else
        lhsElements.put(element, Character.getNumericValue(
    }
}
}
returnMapList.add(lhsElements)

for(int i=0; i<rhsMolecules.size(); i++) {
    molecule = rhsMolecules.get(i);
    firstChar = molecule.charAt(0);
    if(firstChar >='0' && firstChar <='9')
        molecule = molecule.substring(1);

    for(int j=0; j<molecule.length(); j++) {
        if(Character.isLowerCase(molecule.charAt(j))) {
            element = molecule.substring(j-1, j+1);
            if(rhsElements.containsKey(element))
                rhsElements.put(element, rhsElements.get
                    (element) + Character.getNumeric
                    (firstChar));
            else
                rhsElements.put(element, Character.getNumericValue
                    (firstChar));
        }
    }
}
```

```
        return MapList.add(rhsElements);
        return returnMapList;
    }
}
public class Generalized {
    public static void main(String args[]) {
        Scanner keyboard = new Scanner(System.in);
        String eqn;
        eqn = keyboard.nextLine();
        ChemicalEquation equation = new ChemicalEquation(eqn);
        ArrayList<ArrayList<String>> allMolecules
                = equation.getMolecules();
        equation.printNumOfMolecules(allMolecules.get(0), "LHS");
        equation.printNumOfMolecules(allMolecules.get(1), "RHS");
        ArrayList<Map<String, Integer>> countList = equation.
                getElementCount(allMolecules.get(0), allMolecules.get(1));
        Map<String, Integer> lhsCount = countList.get(0);
                // rhsCount = countList.get(1);
        Set<String> keys = lhsCount.keySet();
        System.out.println("LHS elements");
        for (String key: lhsCount.keySet()) {
            System.out.println(key + " " + lhsCount.get(key));
        }
```



Outputs:

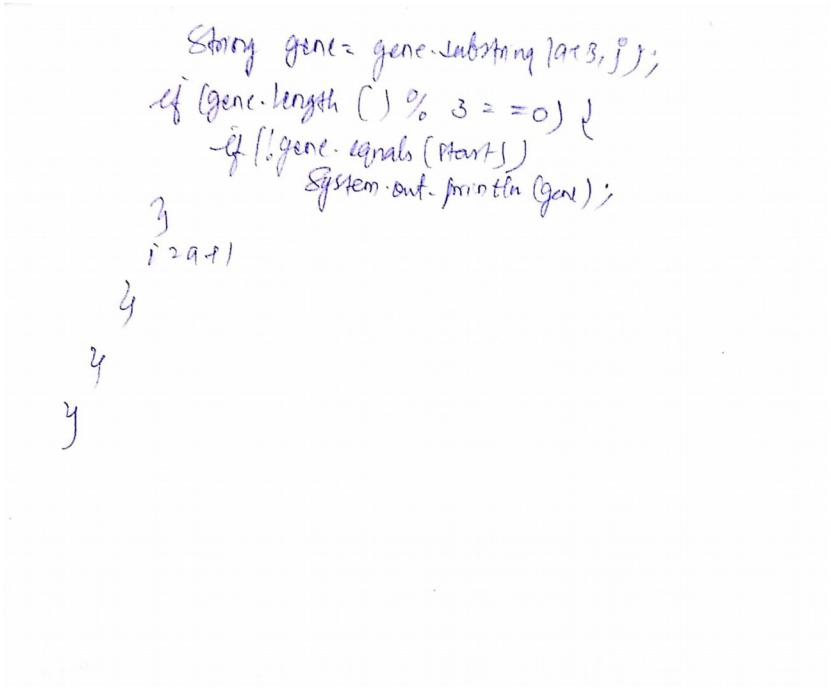
```
krish-thorcode@kkm-ubuntu:~/My_files/College_related_stuffs/Fall_2018-19/Courses
/Advanced JAVA/Lab/challenging_exercises/question3$ java Generalized
2NaOH + H2SO4 -> Na2SO4 + 2H2O
Number of molecules LHS:
NaOH- 2
H2SO4- 1
Number of molecules RHS:
Na2SO4- 1
H2O- 2
LHS elements:
Na 2
S 1
H 4
O 6
RHS elements:
Na 2
S 1
H 4
O 6
The equation is balanced since number(elements in lhs) = number(elements in rhs)
```

Question 4:

Generalized code for all Low + Medium + High levels:

```
Question 4:

class FindGenome {

    public static void main(String args[]) {
        Scanner keyboard = new Scanner(System.in);
        System.out.println("Please Enter the Genome");
        String genome = keyboard.nextLine();
        String start = "ATG";
        String end1 = "TAG";
        String end2 = "TAA";
        String end3 = "TGA";
        int flag = 0;
        char[] valid = genome.toCharArray();
        for(int i=0; i<valid.length; i++) {
            switch(valid[i]) {
                case 'A':
                case 'C':
                case 'G':
                case 'T':
                    break;
                default:
                    flag=1;
                    break;
            }
            if (flag==1) {
                System.out.println("Invalid Input");
                break;
            }
        }
    }
```

```
int i=0; j;

System.out.println("Genes: ");
int a, b, c, d
for (int i=0; i<genome.length; i++) {
    int a = genome.indexOf(start, i);
    if (a==-1)
        break;
    int b = genome.indexOf(end1, a+3);
    int c = genome.indexOf(end2, a+3);
    int d = genome.indexOf(end3, a+3);

if (a==-1)
    break;
b = genome.indexOf(end1, a+3);
c = genome.indexOf(end2, a+3);
  d = genome.indexOf(end3, a+3);

if (b==-1)
    b = genome.length()+30;
if (c==-1)
    c = genome.length()+30;
if (d==-1)
    d = genome.length()+30;

if (b<c && b<d)
    j=b
else if (c<b && c<d)
    j=c;
else j=d;

if (b>genome.length() && d>genome.length() &&
                         c>genome.length())
    break;
```

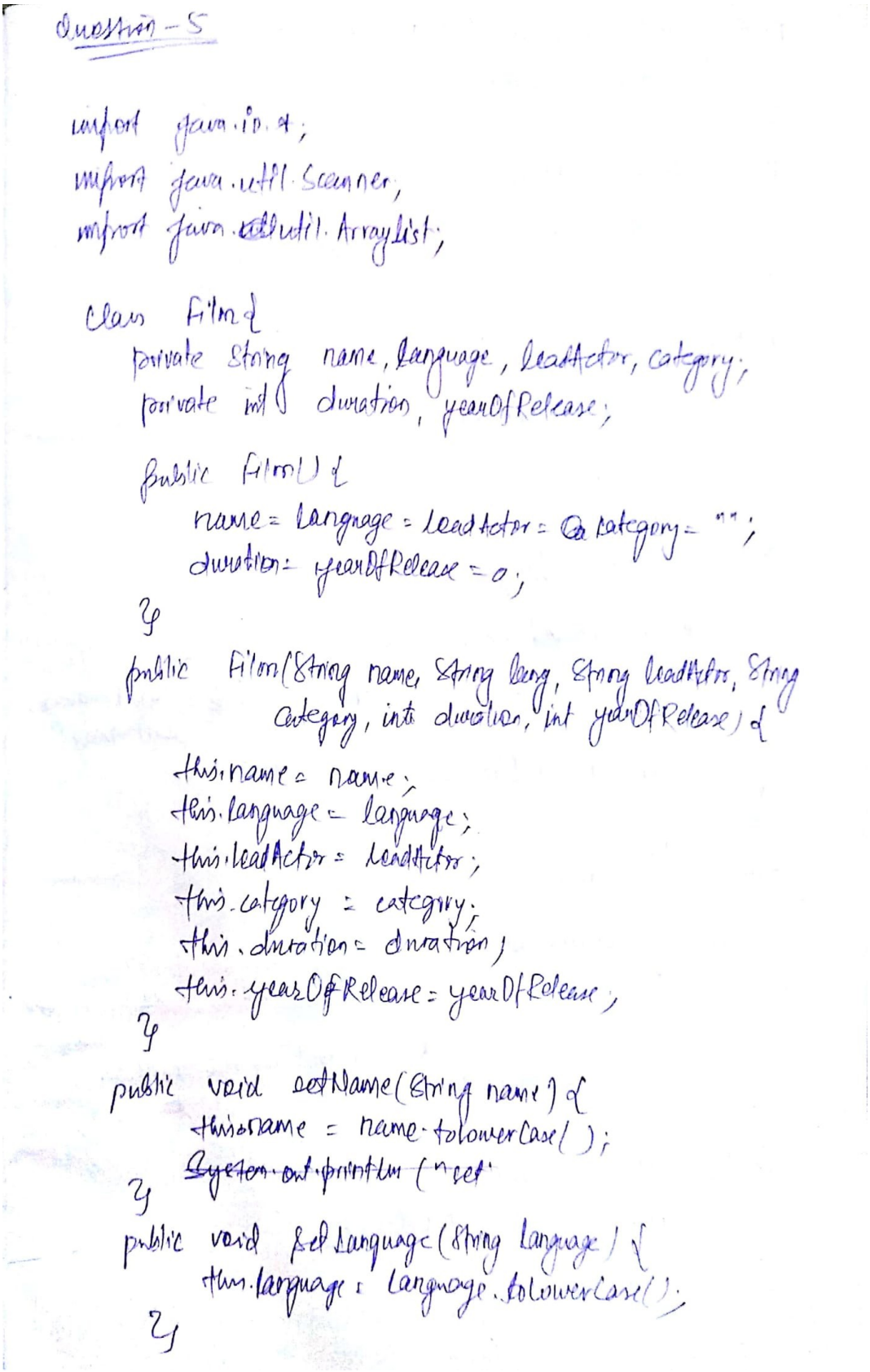


Output:

```
krish-thorcode@kkm-ubuntu:~/My_files/College_related_stuffs/Fall_2018-19/Courses
/Advanced JAVA/Lab/challenging_exercises/question4$ javac Genome.java
krish-thorcode@kkm-ubuntu:~/My_files/College_related_stuffs/Fall_2018-19/Courses
/Advanced JAVA/Lab/challenging_exercises/question4$ java Genome
Enter the genome string:
TTATGTTTTAAGGATGGGGCGTTAGTT
Genes in TTATGTTTTAAGGATGGGGCGTTAGTT:
TTT
GGGCGT
```

Question 5:

Generalized code for all Low + Medium + High levels:



```
Question - 5

import java.io.*;
import java.util.Scanner;
import java.util.ArrayList;

class Film{
    private String name, language, leadActor, category;
    private int duration, yearOfRelease;

    public Film(){
        name = language = leadActor = category = "";
        duration = yearOfRelease = 0;
    }
    public Film(String name, String lang, String leadActor, String
            category, int duration, int yearOfRelease){
        this.name = name;
        this.language = language;
        this.leadActor = leadActor;
        this.category = category;
        this.duration = duration;
        this.yearOfRelease = yearOfRelease;
    }
    public void setName(String name){
        this.name = name.toLowerCase();
        System.out.println("set
    }
    public void setLanguage(String language){
        this.language = language.toLowerCase();
    }
```

```
public void setCategory(String category){
        this.category = category;
}
public void setYearOfRelease(int yearOfRelease){
        this.yearOfRelease = yearOfRelease;

public String getname(){
        String returnName = this.name.substring(0,1).toUpperCase
                + this.name.substring(1);
    return returnName;
public String getLanguage(){
    String returnLanguage = this.language.substring(0,1).
            toUpperCase() + this.Language.
            substring(1);

    return returnLanguage;
}
public String getLeadActor(){
    String returnLeadActor = this.LeadActor.substring(0,1).
            toUpperCase() + this.leadActor.toUpper
            case();
            toUpperCase() + this.leadActor.subst
            (1);

public int getYear(){
    return this.yearOfRelease;
}
}
```

```
class SourceOfFunctions {

    Film[][] create2DArray(Film films[]) {

        Film[][] array2D = new Film[6][films.length];
        int counter[] = new int[6];

        for(Film film: films) {
            if(film.getYearOfRelease() == 1971)
                array2D[0][counter[0]++] = film;
            else if(film.getYearOfRelease() > 1971 &&
                    film.getYearOfRelease() <= 1980)
                array2D[1][counter[1]++] = film;

            else if(film.getYearOfRelease() >= 1981 &&
                    film.getYearOfRelease() <= 1990)
                array2D[2][counter[2]++] = film;

            else if(film.getYearOfRelease() >= 1991 &&
                    film.getYearOfRelease() <= 2000)
                array2D[3][counter[3]++] = film;

            else if(film.getYearOfRelease() > 2001 &&
                    film.getYearOfRelease() <= 2010)
                array2D[4][counter[4]++] = film;

            else array2D[5][counter[5]++] = film;

        }
        return array2D;
    }
```

```
ArrayList<Film> getRajiniFilms(ArrayList<Film> films){
    ArrayList<Film> rajinifilms = new ArrayList<Film>();
    for(Film film: films){
        if((film.getLeadActor().equals("Rajinikanth")||
            film.getLeadActor().equals("Rajini")) &&
            (film.getLanguage().equals("Tamil"))){
                rajinifilms.add(film);
        }
    }
    return rajinifilms;
}

ArrayList<Film> getArnoldFilms(ArrayList<Film> films){
    ArrayList<Film> arnoldfilms = new ArrayList<Film>();
    for(Film film: films){
        if((film.getLeadActor().equals("Arnold")) &&
            (film.getLanguage().equals("English"))){
                arnoldfilms.add(film);
        }
    }
    return Arnoldfilms;
}
```

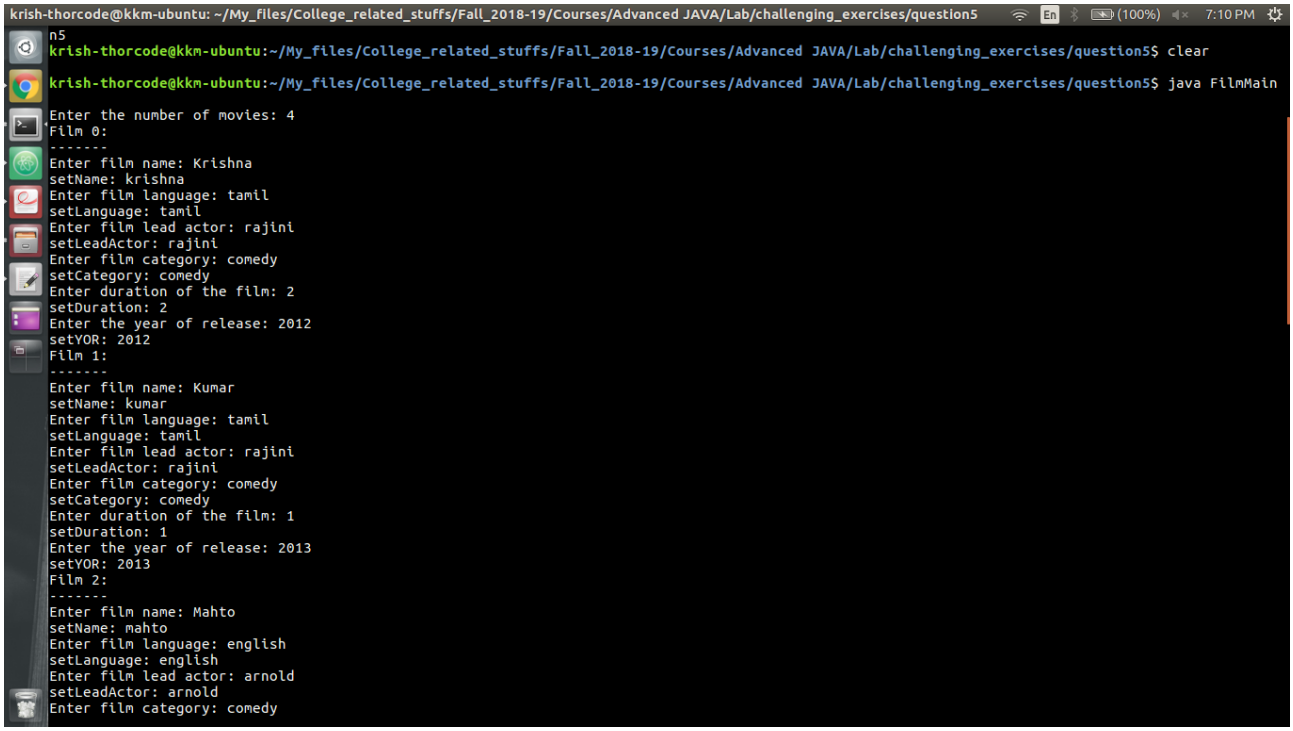
```
ArrayList<Film> getComedyFilms(ArrayList<Film> films, String
                                actor){

>(); ArrayList<Film> comedyFilmsWithActor = new ArrayList<Film>();
     actor = actor.substring(0,1).toUpperCase() + actor.substring(1);
     for(Film film: films){
       if(film.category().equals("Comedy") && film.getLeadActor().
                                              equals(actor))
           comedyFilmsWithActor.add(film);
       }
       return ~~comedyFilmsWithActor~~ comedyFilmsWithActor;
     }

     Film getShortestFilm(ArrayList<Film> films){
       Film shortestFilm = films.get(0);
       ~~for~~
       for(Film film: films){
         if(shortestFilm.getDuration() > film.getDuration())
             shortestFilm = film;
         }
       return shortestFilm;
     }
   }

   class FilmMain{
     public static void main(String args[]) throws IOException{
       Scanner ~~key~~ scanner = new Scanner(System.in);
       BufferedReader br = new BufferedReader(new
                                InputStreamReader(System.in));
       int n;
```

```
System.out.print("Enter the num of movies: ");
n = scanner.nextInt();

ArrayList<Film> films = new ArrayList<Film>();
int numOfFilms = 0;

for (int i=0; i<n; i++) {
    Film film = new Film();
    System.out.println("Film " + i + ":");
    System.out.println("-----");
    System.out.print("Enter film name: ");
    String fname = br.readLine();
    film.setName(fname);
    System.out.print("Enter film lead actor: ");
    String lead = br.readLine();

    System.out.print("Enter film category: ");
    String category = br.readLine();
    film.setCategory(category);

    System.out.print("Enter duration");
    int duration = Integer.parseInt(br.readLine());
    film.setYearOfRelease(y);
    film.setDuration(duration);
}
```

```
SourceOfFunctions sof = new SourceOfFunctions();
ArrayList<Film> rajnifilms = sof.getRajnifilms(films);
ArrayList<Film> arnoldFilms = sof.getArnoldFilms(films);
"               comedyFilms = sof.getComedyFilms(films);

for(Film rajnifilm: rajnifilms){
    System.out.println("Film name: " + rajnifilm.getName());
    System.out.println("Film lang" + rajnifilm.getLang());
    System.out.println("Film actor" + rajnifilm.getLeadActor());
    System.out.println("Film category" + rajnifilm.getCategory());
    System.out.println("Film duration" + rajnifilm.getDuration());
    System.out.println("Film yearOfRel." + rajnifilm.getYOfRelease());
}
// Same for arnoldfilm variable

Film shortestFilm = sof.getShortestFilm(arnoldFilms);
System.out.println("Shortest Arnold film: " + shortestFilm.
                                                getName());

for(Film comedyFilm: comedyFilms){
    System.out.println("Film name: " + comedyFilm.getName());
    System.out.println("Language: " + comedyFilm.getLanguage());
    System.out.println("Actor" + comedyFilm.getActor());
    //Print category, duration, yofrelease
}   // similarly
```

```
System.out.println("Enter the actor to get his comedy
                    Movies: ");
String actor = br.readLine();

ArrayList<Film> actorComedyFilms = lol.getComedyFilms
                                   (films, actor);

for(Film comedyFilm: actorComedyFilms){
    System.out.println("Name: "+comedyFilm.getName());
    System.out.println("Lang: "+comedyFilm.getLang());
    "    "    "  ("Film actor: "+comedyFilm.getActor());
    System.out.println("Category"+comedyFilm.getCategory());
    System.out.println("YearOfRel"+comedyFilm.getYearOfRel());
}
}
}
```

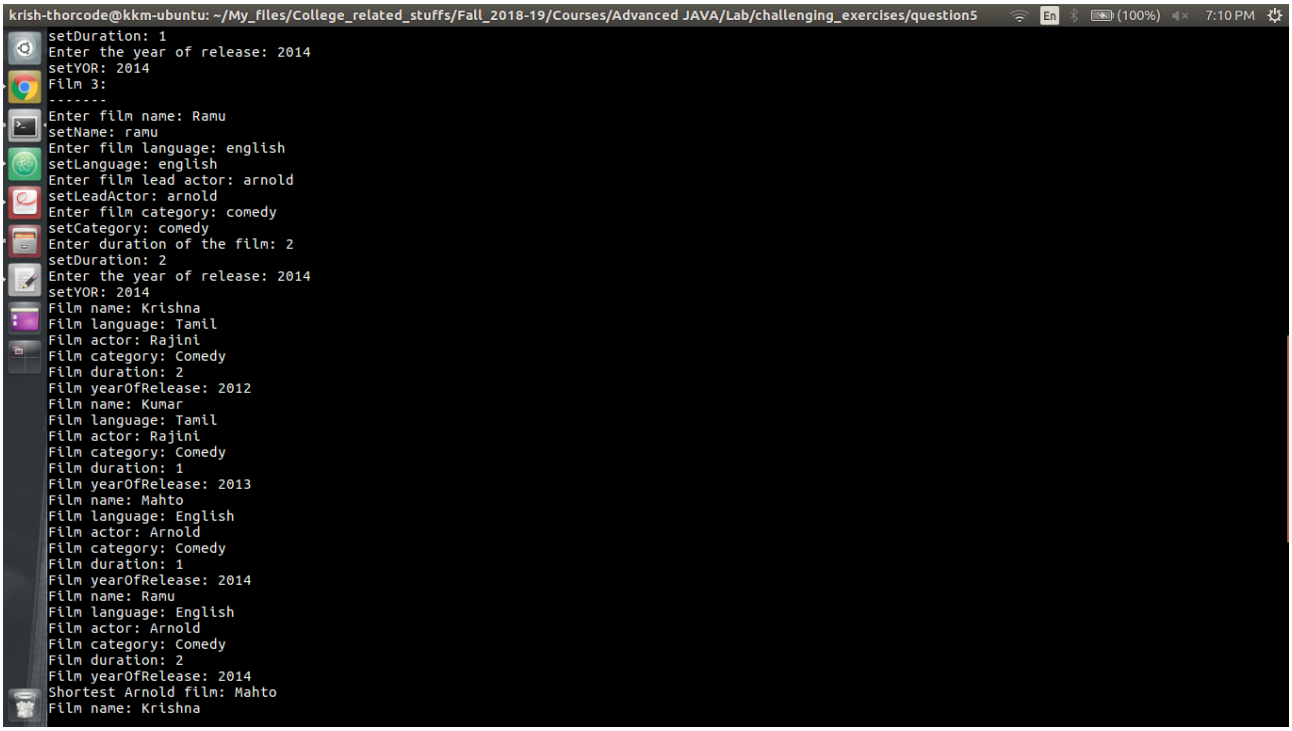
Output:


krish-thorcode@kkm-ubuntu: ~/My_files/College_related_stuffs/Fall_2018-19/Courses/Advanced JAVA/Lab/challenging_exercises/question5
7:10 PM
n5
krish-thorcode@kkm-ubuntu:~/My_files/College_related_stuffs/Fall_2018-19/Courses/Advanced JAVA/Lab/challenging_exercises/question5$ clear
krish-thorcode@kkm-ubuntu:~/My_files/College_related_stuffs/Fall_2018-19/Courses/Advanced JAVA/Lab/challenging_exercises/question5$ java FilmMain
Enter the number of movies: 4
Film 0:

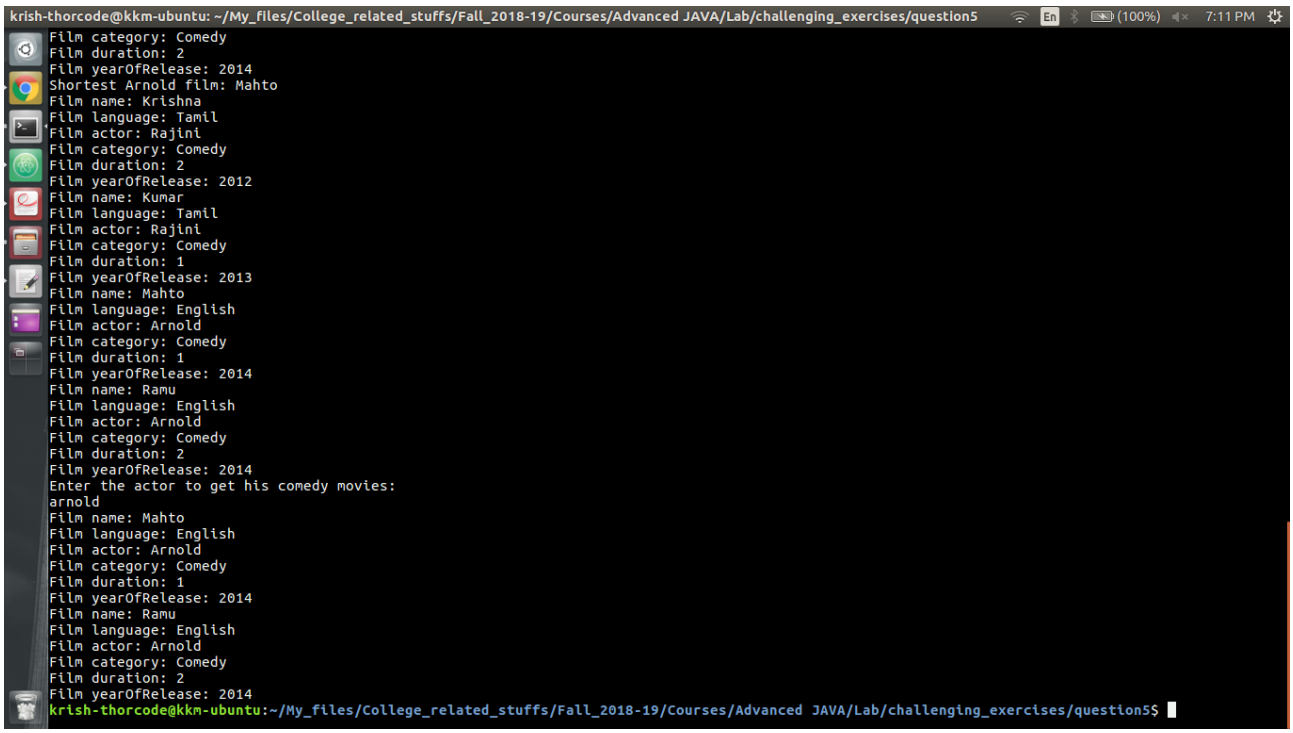
Enter film name: Krishna
setName: krishna
Enter film language: tamil
setLanguage: tamil
Enter film lead actor: rajini
setLeadActor: rajini
Enter film category: comedy
setCategory: comedy
Enter duration of the film: 2
setDuration: 2
Enter the year of release: 2012
setYOR: 2012
Film 1:

Enter film name: Kumar
setName: kumar
Enter film language: tamil
setLanguage: tamil
Enter film lead actor: rajini
setLeadActor: rajini
Enter film category: comedy
setCategory: comedy
Enter duration of the film: 1
setDuration: 1
Enter the year of release: 2013
setYOR: 2013
Film 2:

Enter film name: Mahto
setName: mahto
Enter film language: english
setLanguage: english
Enter film lead actor: arnold
setLeadActor: arnold
Enter film category: comedy


krish-thorcode@kkm-ubuntu: ~/My_files/College_related_stuffs/Fall_2018-19/Courses/Advanced JAVA/Lab/challenging_exercises/question5
7:10 PM
setDuration: 1
Enter the year of release: 2014
setYOR: 2014
Film 3:

Enter film name: Ramu
setName: ramu
Enter film language: english
setLanguage: english
Enter film lead actor: arnold
setLeadActor: arnold
Enter film category: comedy
setCategory: comedy
Enter duration of the film: 2
setDuration: 2
Enter the year of release: 2014
setYOR: 2014
Film name: Krishna
Film language: Tamil
Film actor: Rajini
Film category: Comedy
Film duration: 2
Film yearOfRelease: 2012
Film name: Kumar
Film language: Tamil
Film actor: Rajini
Film category: Comedy
Film duration: 1
Film yearOfRelease: 2013
Film name: Mahto
Film language: English
Film actor: Arnold
Film category: Comedy
Film duration: 1
Film yearOfRelease: 2014
Film name: Ramu
Film language: English
Film actor: Arnold
Film category: Comedy
Film duration: 2
Film yearOfRelease: 2014
Shortest Arnold film: Mahto
Film name: Krishna


krish-thorcode@kkm-ubuntu: ~/My_files/College_related_stuffs/Fall_2018-19/Courses/Advanced JAVA/Lab/challenging_exercises/question5
En
(100%)
7:11 PM
Film category: Comedy
Film duration: 2
Film yearOfRelease: 2014
Shortest Arnold film: Mahto
Film name: Krishna
Film language: Tamil
Film actor: Rajini
Film category: Comedy
Film duration: 2
Film yearOfRelease: 2012
Film name: Kumar
Film language: Tamil
Film actor: Rajini
Film category: Comedy
Film duration: 1
Film yearOfRelease: 2013
Film name: Mahto
Film language: English
Film actor: Arnold
Film category: Comedy
Film duration: 1
Film yearOfRelease: 2014
Film name: Ramu
Film language: English
Film actor: Arnold
Film category: Comedy
Film duration: 2
Film yearOfRelease: 2014
Enter the actor to get his comedy movies:
arnold
Film name: Mahto
Film language: English
Film actor: Arnold
Film category: Comedy
Film duration: 1
Film yearOfRelease: 2014
Film name: Ramu
Film language: English
Film actor: Arnold
Film category: Comedy
Film duration: 2
Film yearOfRelease: 2014
krish-thorcode@kkm-ubuntu:~/My_files/College_related_stuffs/Fall_2018-19/Courses/Advanced JAVA/Lab/challenging_exercises/question5$