Question 1

/\*

Question 1

\*/

import java.util.Map;

import java.util.Scanner;

import java.util.Set;

import java.util.TreeMap;

import java.util.Map.Entry;

import com.sun.corba.se.impl.encoding.OSFCodeSetRegistry.Entry;

import org.omg.SendingContext.RunTime;

/\*\*

   A class to keep track of students and their course grades.

\*/

public class grades\_question1

{

   /\*\*

Prints the options menu and returns the letter that the user types does not check for invalid selection

      @return the next token on the input stream

   \*/

   public static String printMenuAndGetChoice()

   {

      System.out.println("A)dd student.");

      System.out.println("R)emove student.");

      System.out.println("M)odify grades.");

      System.out.println("P)rint all grades.");

      System.out.println("Q)uit.");

      System.out.println("Select one: ");

      Scanner in = new Scanner(System.in);

      return in.next().toUpperCase();

   }

   /\*\*

      Prints the students and grades

      @param gradeMap the map to print

   \*/

   public static void printGrades(Map<String, String> gradeMap)

   {

      Set<String> studentSet = gradeMap.keySet();

      TreeMap<String, String> sorted = new TreeMap<>(gradeMap);

      Set<Entry<String, String>> mappings = sorted.entrySet();

      for(Entry<String, String> mapping : mappings){

        System.out.println("\*"+ mapping.getKey() +" "+ mapping.getValue());

      }

      // TO do, write code to print student and their grades

   }

   /\*\*

Modifies an entry based on user input.  Prints an error if an invalid student is modified

      @param gradeMap the map to modify

   \*/

   public static void modifiyStudent(Map<String, String> gradeMap)

   {

      Scanner in = new Scanner(System.in);

      System.out.println("Enter the name of the student to modify: ");

      String name = in.next();

      if (gradeMap.containsKey(name))

      {

         System.out.println("Enter the new grade: ");

         String grade = in.next();

         gradeMap.put(name,grade);

         printGrades(gradeMap);

         // To Do; write code to modify stdeutns

      }

      else

      {

         System.out.println("No such student exists!");

      }

   }

   /\*\*

      Removes a student from the map based on user input

      @param gradeMap the map to remove the student from

   \*/

   public static void removeStudent(Map<String, String> gradeMap)

   {

      Scanner in = new Scanner(System.in);

      System.out.println("Enter the name of the student to remove: ");

      String name = in.nextLine();

      if( gradeMap.containsKey(name))

      {

          gradeMap.remove(name);

          System.out.println("The student has been removed!");

      }

      else

      {

          System.out.println("The student doesnt not exist");

      }

     //TO do; write code to remove student

   }

   /\*\*

      Adds a student based on user input.  Prints an error if a student

      is added that already exists in the map.

      @param gradeMap the map to add the student to

   \*/

   public static void addStudent(Map<String, String> gradeMap)

   {

      Scanner in = new Scanner(System.in);

      System.out.println("Enter the name of the student to add: ");

      String name = in.nextLine();

      if (!gradeMap.containsKey(name))

      {

         System.out.println("Enter the new grade: ");

         String grade = in.nextLine();

         gradeMap.put(name, grade);

         printGrades(gradeMap);

         // To Do; write code to add stduent and its grade to map

      }

      else

      {

         System.out.println("That student is already in the roster.");

      }

   }

   public static void main(String[] args)

   {

      Map<String, String> gradeMap = new TreeMap<String, String>();

      String choice = printMenuAndGetChoice();

      while (!choice.equals("Q"))

      {

         if (choice.equals("A"))

         {

            addStudent(gradeMap);

         }

         if (choice.equals("R"))

         {

            removeStudent(gradeMap);

         }

         if (choice.equals("M"))

         {

            modifiyStudent(gradeMap);

         }

         if (choice.equals("P"))

         {

            printGrades(gradeMap);

         }

         choice = printMenuAndGetChoice();

      }

   }

}

The Output

bash-3.2$ /Library/Java/JavaVirtualMachines/jdk-9.jdk/Contents/Home/bin/java -Dfile.encoding=UTF-8 -cp "/Users/nsing048/Library/Application Support/Code/User/workspaceStorage/5a35887b43febcaa1cefbb96b492755b/redhat.java/jdt\_ws/jdt.ls-java-project/bin" grades\_question1

A)dd student.

R)emove student.

M)odify grades.

P)rint all grades.

Q)uit.

Select one:

a

Enter the name of the student to add:

jay

Enter the new grade:

A

\*jay A

A)dd student.

R)emove student.

M)odify grades.

P)rint all grades.

Q)uit.

Select one:

a

Enter the name of the student to add:

neetu

Enter the new grade:

A

\*jay A

\*neetu A

A)dd student.

R)emove student.

M)odify grades.

P)rint all grades.

Q)uit.

Select one:

r

Enter the name of the student to remove:

neetu

The student has been removed!

A)dd student.

R)emove student.

M)odify grades.

P)rint all grades.

Q)uit.

Select one:

m

Enter the name of the student to modify:

jay

Enter the new grade:

B

\*jay B

A)dd student.

R)emove student.

M)odify grades.

P)rint all grades.

Q)uit.

Select one:

p

\*jay B

A)dd student.

R)emove student.

M)odify grades.

P)rint all grades.

Q)uit.

Select one:

a

Enter the name of the student to add:

neetu

Enter the new grade:

A

\*jay B

\*neetu A

A)dd student.

R)emove student.

M)odify grades.

P)rint all grades.

Q)uit.

Select one:

a

Enter the name of the student to add:

andrew

Enter the new grade:

B

\*andrew B

\*jay B

\*neetu A

A)dd student.

R)emove student.

M)odify grades.

P)rint all grades.

Q)uit.

Select one:

p

\*andrew B

\*jay B

\*neetu A

A)dd student.

R)emove student.

M)odify grades.

P)rint all grades.

Q)uit.

Select one:

Question 2

// Question 2

import java.util.Map;

import java.util.Scanner;

import java.util.Set;

import java.util.TreeMap;

/\*\*

   A class to keep track of students and their course grades.

\*/

public class Grades

{

   /\*\*

Prints the options menu and returns the letter that the user types does not check for invalid selection

      @return the next token on the input stream

   \*/

   public static String printMenuAndGetChoice()

   {

      System.out.println("A)dd student.");

      System.out.println("R)emove student.");

      System.out.println("M)odify grades.");

      System.out.println("P)rint all grades.");

      System.out.println("Q)uit.");

      System.out.println("Select one: ");

      Scanner in = new Scanner(System.in);

      return in.next().toUpperCase();

   }

   /\*\*

      Prints the students and grades

      @param gradeMap the map to print

   \*/

   public static void printGrades(Map<Student, String> gradeMap)

   {

      Set<Student> studentSet = gradeMap.keySet();

      for (Student student : studentSet)

      {

         System.out.printf("%s: %s\n", student, gradeMap.get(student));

      }

   }

   /\*\*

Modifies an entry based on user input.  Prints an error if an invalid student is modified

      @param gradeMap the map to modify

      @param idToStudentMap  the map to associate student id with a student

   \*/

   public static void modifiyStudent(Map<Student, String> gradeMap,

      Map<Integer, Student> idToStudentMap)

   {

      Scanner in = new Scanner(System.in);

      System.out.println("Enter the ID of the student to modify: ");

      int id = in.nextInt();

      if (idToStudentMap.containsKey(id))

      {

         System.out.println("Enter the new grade: ");

         String grade = in.next();

         gradeMap.put(idToStudentMap.get(id), grade);

      }

      else

      {

         System.out.println("No such student exists!");

      }

   }

   /\*\*

      Removes a student from the map based on user input

      @param gradeMap the map to remove the student from

      @param idToStudentMap the map to associate student id with a student

   \*/

   public static void removeStudent(Map<Student, String> gradeMap,

      Map<Integer, Student> idToStudentMap)

   {

      Scanner in = new Scanner(System.in);

      System.out.println("Enter the id of the student to remove: ");

      int id = in.nextInt();

      //TO Do; write code to remove student; make sure to remove stdeunt from both map, id map and grade map

      gradeMap.remove(idToStudentMap.get(id));

      idToStudentMap.remove(id);

   }

   /\*\*

      Adds a student based on user input.  Prints an error if a student

      is added that already exists in the map.

      @param idToGradeMap the map to add the student to

      @param idToStudentMap the map to associate student id with a student

   \*/

   public static void addStudent(Map<Student, String> idToGradeMap,

      Map<Integer, Student> idToStudentMap)

   {

      Scanner in = new Scanner(System.in);

      Student student = Student.createStudent();

      if (!idToStudentMap.containsKey(student.getId()))

      {

         System.out.println("Enter the new grade: ");

         // TO Do; write code to add student

         String grade = in.next();

         idToGradeMap.put(student, grade);

         idToStudentMap.put(student.getId(), student);

      }

      else

      {

         System.out.println("That student is already in the roster.");

      }

   }

   public static void main(String[] args)

   {

      Map<Student, String> gradeMap = new TreeMap<Student, String>();

      Map<Integer, Student> idToStudentMap = new TreeMap<Integer, Student>();

      String choice = printMenuAndGetChoice();

      while (!choice.equals("Q"))

      {

         if (choice.equals("A"))

         {

            addStudent(gradeMap, idToStudentMap);

         }

         if (choice.equals("R"))

         {

            removeStudent(gradeMap, idToStudentMap);

         }

         if (choice.equals("M"))

         {

            modifiyStudent(gradeMap, idToStudentMap);

         }

         if (choice.equals("P"))

         {

            printGrades(gradeMap);

         }

         choice = printMenuAndGetChoice();

      }

   }

}

// Student class for question 2

import java.util.Scanner;

/\*\*

   A class to represent a student.

\*/

public class Student implements Comparable<Student>

{

   private String firstName;

   private String lastName;

   private int id;

   /\*\*

      The id getter

      @return the id

   \*/

   public int getId()

   {

      return id;

   }

   /\*\*

      Constructs a new student with a given first and last name, and id

      @param firstName

      @param lastName

      @param id

   \*/

   public Student(String firstName, String lastName, int id)

   {

      //TO Do; complete the Constructor

      this.firstName = firstName;

      this.lastName = lastName;

      this.id = id;

   }

   public Student()

   {

      this.firstName = null;

      this.lastName = null;

      this.id = 0;

   }

   /\*\*

Compares students by sorting by last name. If last names are equal, then by first name, if first names are identical, then by id.

      @param other the other student to compare with

   \*/

   public int compareTo(Student other)

   {

      int lastNameCompare = this.lastName.compareTo(other.lastName);

      if (lastNameCompare != 0)

      {

         return lastNameCompare;

      }

      int firstNameCompare = this.firstName.compareTo(other.firstName);

      //TO DO; write code to compare firs name

      if(firstNameCompare != 0)

      {

         return firstNameCompare;

      }

      return this.id - other.id;

   }

   /\*\*

      Determines equality based on string values

   \*/

   public boolean equals(Object obj)

   {

      return this.compareTo((Student) obj) == 0;

   }

   /\*\*

      Generates a string of the student in the form: lastname, firstname id

   \*/

   public String toString()

   {

      return lastName + ", " + firstName + " " + id;

   }

   /\*\*

      Creates a new student based on user input

      @return a new student with values defined by user input

    \*/

   public static Student createStudent()

   {

      Scanner in = new Scanner(System.in);

      System.out.println("Please enter first name: ");

      String first = in.nextLine();

      // TO Do; write code to assign first name

      System.out.println("Please enter last name: ");

      // TO Do; write code to assign last name

      String last = in.nextLine();

      System.out.println("Please enter ID: ");

      // TO Do; write code to assign id

      int id = in.nextInt();

      return new Student(first, last, id);

   }

}

OutPut

bash-3.2$ /Library/Java/JavaVirtualMachines/jdk-9.jdk/Contents/Home/bin/java -Dfile.encoding=UTF-8 -cp "/Users/nsing048/Library/Application Support/Code/User/workspaceStorage/5a35887b43febcaa1cefbb96b492755b/redhat.java/jdt\_ws/jdt.ls-java-project/bin" Grades

A)dd student.

R)emove student.

M)odify grades.

P)rint all grades.

Q)uit.

Select one:

a

Please enter first name:

neetu

Please enter last name:

singh

Please enter ID:

123

Enter the new grade:

A

A)dd student.

R)emove student.

M)odify grades.

P)rint all grades.

Q)uit.

Select one:

a

Please enter first name:

Jay

Please enter last name:

Kim

Please enter ID:

124

Enter the new grade:

A

A)dd student.

R)emove student.

M)odify grades.

P)rint all grades.

Q)uit.

Select one:

a

Please enter first name:

Andrew

Please enter last name:

Rettko

Please enter ID:

125

Enter the new grade:

bash-3.2$ cd /Users/nsing048/Desktop/AdvData ; /Library/Java/JavaVirtualMachines/jdk-9.jdk/Contents/Home/bin/java -Dfile.encoding=UTF-8 -cp "/Users/nsing048/Librar

y/Application Support/Code/User/workspaceStorage/5a35887b43febcaa1cefbb96b492755b/redhat.java/jdt\_ws/jdt.ls-java-project/bin" Grades

A)dd student.

R)emove student.

M)odify grades.

P)rint all grades.

Q)uit.

Select one:

a

Please enter first name:

neetu

Please enter last name:

singh

Please enter ID:

1

Enter the new grade:

A

A)dd student.

R)emove student.

M)odify grades.

P)rint all grades.

Q)uit.

Select one:

a

Please enter first name:

jay

Please enter last name:

kim

Please enter ID:

2

Enter the new grade:

A

A)dd student.

R)emove student.

M)odify grades.

P)rint all grades.

Q)uit.

Select one:

a

Please enter first name:

andrew

Please enter last name:

rettko

Please enter ID:

bash-3.2$ cd /Users/nsing048/Desktop/AdvData ; /Library/Java/JavaVirtualMachines/jdk-9.jdk/Contents/Home/bin/java -Dfile.encoding=UTF-8 -cp "/Users/nsing048/Librar

y/Application Support/Code/User/workspaceStorage/5a35887b43febcaa1cefbb96b492755b/redhat.java/jdt\_ws/jdt.ls-java-project/bin" Grades

A)dd student.

R)emove student.

M)odify grades.

P)rint all grades.

Q)uit.

Select one:

a

Please enter first name:

neetu

Please enter last name:

singh

Please enter ID:

1

Enter the new grade:

A

A)dd student.

R)emove student.

M)odify grades.

P)rint all grades.

Q)uit.

Select one:

a

Please enter first name:

jay

Please enter last name:

kim

Please enter ID:

2

Enter the new grade:

A

A)dd student.

R)emove student.

M)odify grades.

P)rint all grades.

Q)uit.

Select one:

a

Please enter first name:

andrew

Please enter last name:

rettko

Please enter ID:

3

Enter the new grade:

A

A)dd student.

R)emove student.

M)odify grades.

P)rint all grades.

Q)uit.

Select one:

r

Enter the id of the student to remove:

1

A)dd student.

R)emove student.

M)odify grades.

P)rint all grades.

Q)uit.

Select one:

p

kim, jay 2: A

rettko, andrew 3: A

A)dd student.

R)emove student.

M)odify grades.

P)rint all grades.

Q)uit.

Select one:

m

Enter the ID of the student to modify:

2

Enter the new grade:

A+

A)dd student.

R)emove student.

M)odify grades.

P)rint all grades.

Q)uit.

Select one:

p

kim, jay 2: A+

rettko, andrew 3: A

A)dd student.

R)emove student.

M)odify grades.

P)rint all grades.

Q)uit.

Select one:

Question 3

// Question 3

import java.util.Iterator;

import java.util.List;

import java.util.ArrayList;;

public class removeInRange1

{

    public static void removeInRange(List<Integer> list, int value, int min, int max)

    {

        Iterator<Integer> itr = list.iterator();

        //to do

        int index = 0;

        while(itr.hasNext())

        {

            if(index< max && index >= min )

            {

                if (itr.next() == value)

                {

                    itr.remove();

                    index--;

                    max--;

                }

            }

            else

            {

                itr.next();

            }

            index++;

        }

    }

    public static void main(String []args) {

        List<Integer> list1 = new ArrayList<Integer>();

        list1.add(1);

        list1.add(2);

        list1.add(3);

        list1.add(4);

        list1.add(5);

        list1.add(5);

        list1.add(6);

        list1.add(5);

        list1.add(7);

        list1.add(5);

        list1.add(5);

        list1.add(8);

        list1.add(9);

        list1.add(5);

        list1.add(0);

        list1.add(2);

        list1.add(3);

        list1.add(5);

        list1.add(5);

        list1.add(5);

        list1.add(5);

        System.out.println(list1);

        removeInRange(list1, 5, 0, 21);

        System.out.println("All the 5 are gone");

        System.out.println(list1);

    }

}

bash-3.2$ /Library/Java/JavaVirtualMachines/jdk-9.jdk/Contents/Home/bin/java -Dfile.encoding=UTF-8 -cp "/Users/nsing048/Library/Application Support/Code/User/workspaceStorage/5a35887b43febcaa1cefbb96b492755b/redhat.java/jdt\_ws/jdt.ls-java-project/bin" removeInRange1

[1, 2, 3, 4, 5, 5, 6, 5, 7, 5, 5, 8, 9, 5, 0, 2, 3, 5, 5, 5, 5]

All the 5 are gone

[1, 2, 3, 4, 6, 7, 8, 9, 0, 2, 3]

bash-3.2$

Question 4

// Question 4

import java.util.ArrayList;

import java.util.Iterator;

import java.util.List;

public class CUS1151\_\_lastName\_HW1

{

    public static void main(String[] args)

    {

        List<Integer> list1 = new ArrayList<>();

        list1.add(1);

        list1.add(2);

        list1.add(3);

        System.out.println(list1);

        List<Integer> list2 = new ArrayList<>();

        list2.add(4);

        list2.add(5);

        list2.add(6);

        System.out.println(list2);

        System.out.println(alternate(list1, list2));

    }

         public static List <Integer> alternate(List<Integer> list1, List<Integer> list2)

            {

            List result = new ArrayList<Integer>(); // we have a arraylist of integers

            Iterator i1 = list1.iterator();

            Iterator i2 = list2.iterator();

        // To do return result

            while(i1.hasNext() || i2.hasNext())

            {

            // add first number from list1 to arraylist

                if(i1.hasNext())

                {

                    result.add(i1.next());

                }

            // add the next number from list2 to arraylist

                if(i2.hasNext())

                {

                    result.add(i2.next());

                }

            }

        // return the arraylist

              return result;

        }

    }

The output

bash-3.2$ /Library/Java/JavaVirtualMachines/jdk-9.jdk/Contents/Home/bin/java -Dfile.encoding=UTF-8 -cp "/Users/nsing048/Library/Application Support/Code/User/workspaceStorage/5a35887b43febcaa1cefbb96b492755b/redhat.java/jdt\_ws/jdt.ls-java-project/bin" CUS1151\_\_lastName\_HW1

[1, 2, 3]

[4, 5, 6]

[1, 4, 2, 5, 3, 6]

bash-3.2$