Doctor Details(---RETIRED---)

Grade settings: Maximum grade: 100
Disable external file upload, paste and drop external content: Yes
Based on: Doctor Details(---RETIRED---)
Run: Yes **Evaluate**: Yes
Automatic grade: Yes

Aims Hospital wants to automate the details of doctors in their hospital. They would like to automate the process of extracting doctors data. You have been approached to develop a Java program for their requirements.

Component Specification: DoctorMain

Type (Class)	**Attributes**	**Methods**
DoctorMain	Map<String, String> doctorDetailsMap	The getter and setter methods for the attribute are included in the code skeleton.

Note: *Here the doctorDetailsMap , holds the Key as doctorId and Value as specialization.*

Requirement 1: Find the specialization based on the given doctorId.

Type (Class)	**Methods**	**Responsibilities**
DoctorMain	public String **findSpecialization**(String doctorId)	This method accepts a **doctorId** as an argument, and if the given doctorId matches the doctorIds present in the map, it must return the specialisation of the given doctorId. Else return a string **"Invalid doctor id".** ***Condition:*** *doctorId is case-sensitive.*

Requirement 2: Filter the doctors based on the given specialization

Type (Class)	Methods	Responsibilities
DoctorMain	public List<String> **findDoctorsBasedOnTheGivenSpecialization**(String specialization)	This method accepts specialization as an argument, filters the doctor Ids based on the given specialisation, and returns the list of doctorId's. ***Condition:*** *specialization is case-insensitive.*

You are provided with the main method as code template and it is excluded from evaluation.

Note:

- In the Sample Input / Output provided, the highlighted text in bold corresponds to the input given by the user, and the rest of the text represents the output.
- Ensure to follow the object-oriented specifications provided in the question description.
- Ensure to provide the names for the classes, attributes, and methods as specified in the question description.
- Adhere to the code template, if provided.

Sample Input/Output 1

Enter number of records to be added:

5

Enter the details (doctorId : specialization):

HAR101:Neurology

HAR102:Endocrinology

HAR103:Dermatology

HAR104:Cardiology

HAR105:endocrinology

Enter the doctor id to be searched

HAR104

The given doctor HAR104 is specialized in Cardiology

Enter the specialization to be searched

Endocrinology

Doctors specialized in Endocrinology are

HAR105

HAR102

Sample Input/Output 2

Enter number of records to be added:

4

Enter the details (doctorId : specialization):

HAR101:Radiology

HAR102:Neurology

HAR103:Oncology

HAR104:Neurology

Enter the doctor id to be searched

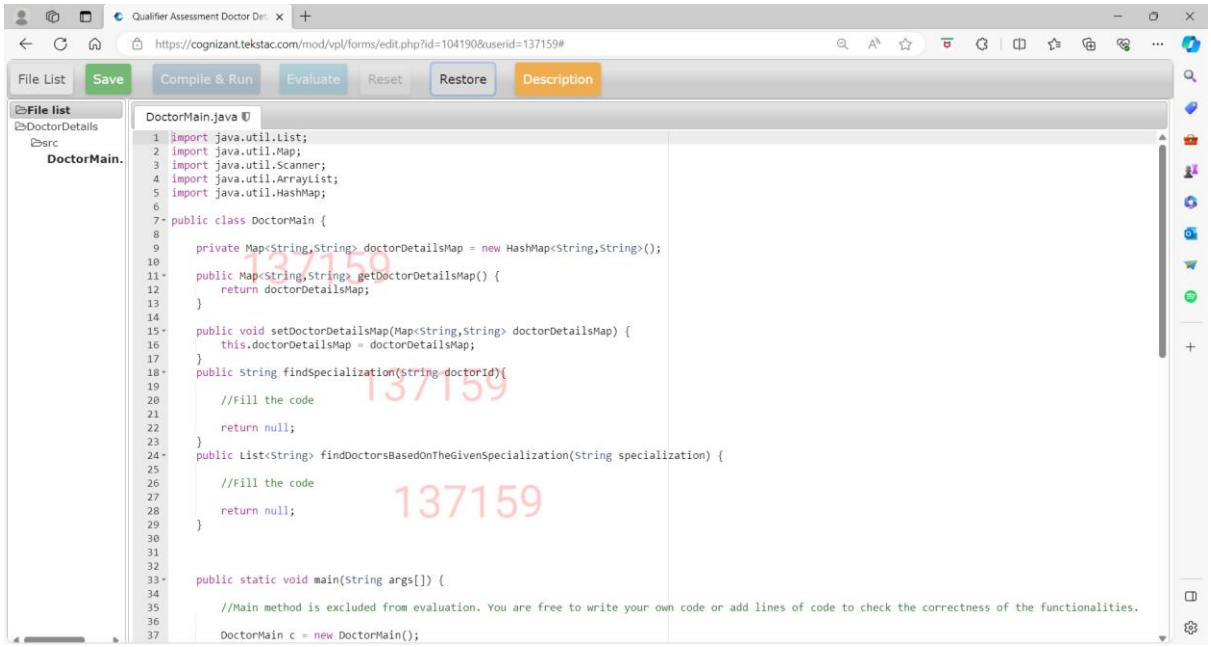
HAR107

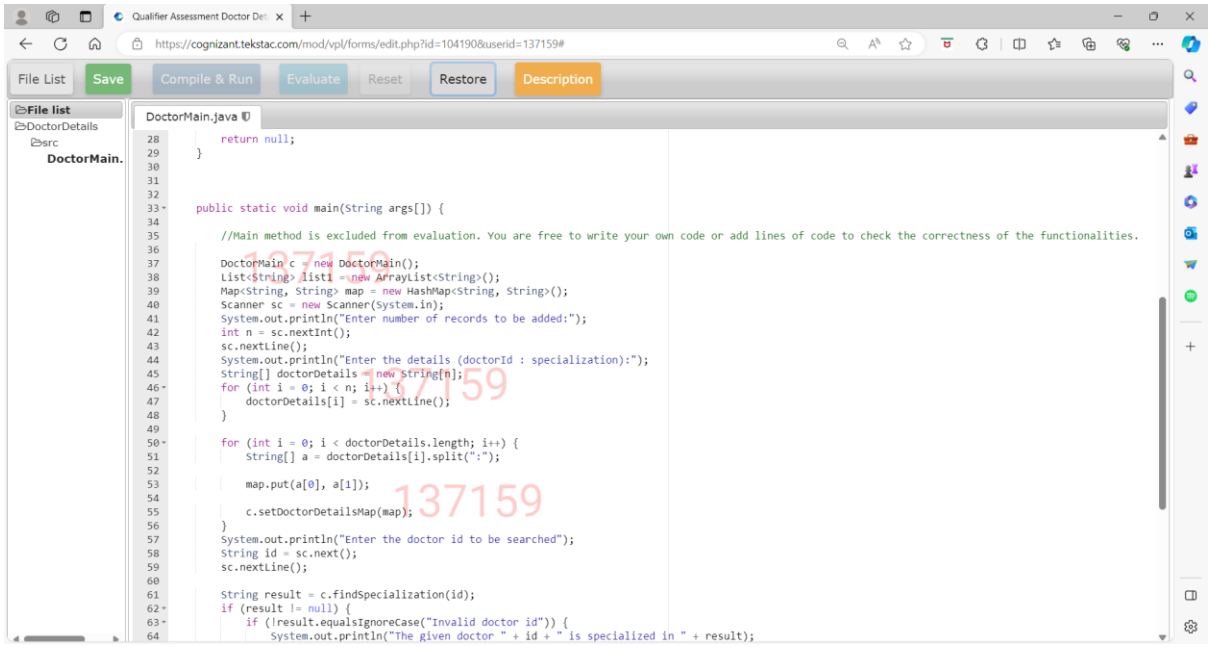
Invalid doctor id

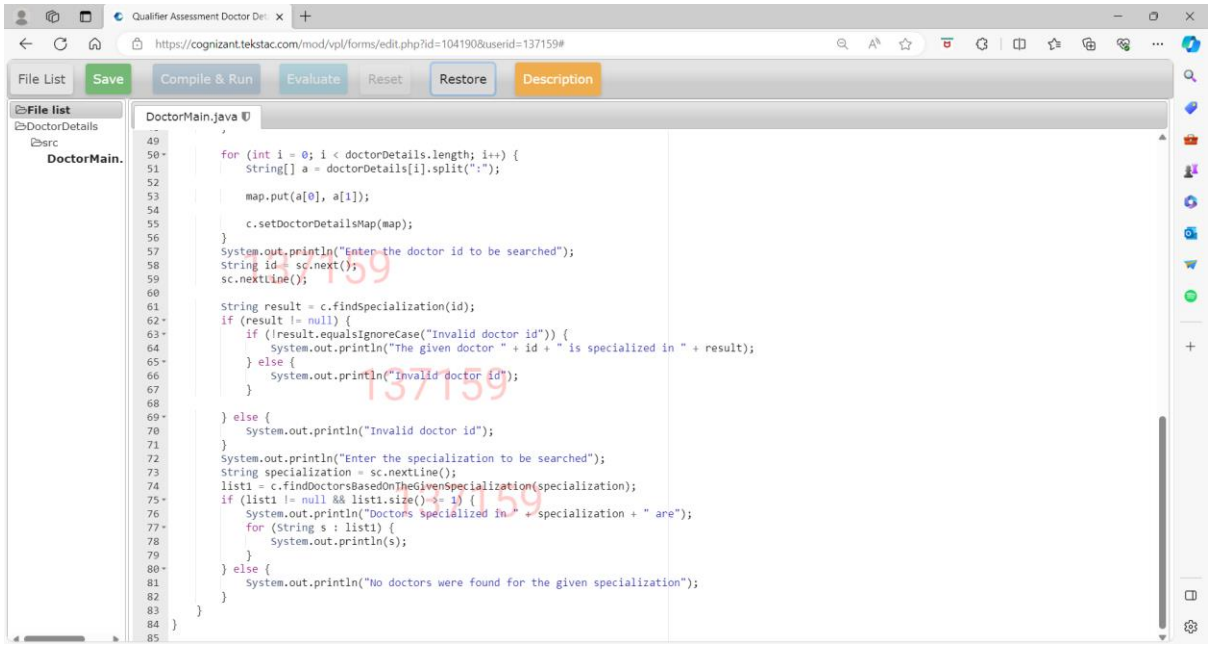
Enter the specialization to be searched

Pediatrics

No doctors were found for the given specialization


Qualifier Assessment Doctor De
https://cognizant.tekstac.com/mod/vpl/forms/edit.php?id=104190&userid=137159#
File List
Save
Compile & Run
Evaluate
Reset
Restore
Description
File list
DoctorDetails
src
DoctorMain.
DoctorMain.java
import java.util.List;
import java.util.Map;
import java.util.Scanner;
import java.util.ArrayList;
import java.util.HashMap;
public class DoctorMain {
private Map<String,String> doctorDetailsMap = new HashMap<String,String>();
public Map<String,String> getDoctorDetailsMap() {
return doctorDetailsMap;
}
public void setDoctorDetailsMap(Map<String,String> doctorDetailsMap) {
this.doctorDetailsMap = doctorDetailsMap;
}
public String findSpecialization(String doctorId){
//Fill the code
return null;
}
public List<String> findDoctorsBasedOnTheGivenSpecialization(String specialization) {
//Fill the code
return null;
}
public static void main(String args[]) {
//Main method is excluded from evaluation. You are free to write your own code or add lines of code to check the correctness of the functionalities.
DoctorMain c = new DoctorMain();


Qualifier Assessment Doctor De
https://cognizant.tekstac.com/mod/vpl/forms/edit.php?id=104190&userid=137159#
File List
Save
Compile & Run
Evaluate
Reset
Restore
Description
File list
DoctorDetails
src
DoctorMain.
DoctorMain.java
return null;
}
public static void main(String args[]) {
//Main method is excluded from evaluation. You are free to write your own code or add lines of code to check the correctness of the functionalities.
DoctorMain c = new DoctorMain();
List<String> list1 = new ArrayList<String>();
Map<String, String> map = new HashMap<String, String>();
Scanner sc = new Scanner(System.in);
System.out.println("Enter number of records to be added:");
int n = sc.nextInt();
sc.nextLine();
System.out.println("Enter the details (doctorId : specialization):");
String[] doctorDetails = new String[n];
for (int i = 0; i < n; i++) {
doctorDetails[i] = sc.nextLine();
}
for (int i = 0; i < doctorDetails.length; i++) {
String[] a = doctorDetails[i].split(":");
map.put(a[0], a[1]);
c.setDoctorDetailsMap(map);
}
System.out.println("Enter the doctor id to be searched");
String id = sc.next();
sc.nextLine();
String result = c.findSpecialization(id);
if (result != null) {
if (!result.equalsIgnoreCase("Invalid doctor id")) {
System.out.println("The given doctor " + id + " is specialized in " + result);


Qualifier Assessment Doctor De
https://cognizant.tekstac.com/mod/vpl/forms/edit.php?id=1041908&userid=137159#
File List
Save
Compile & Run
Evaluate
Reset
Restore
Description
File list
DoctorDetails
src
DoctorMain.
DoctorMain.java
for (int i = 0; i < doctorDetails.length; i++) {
String[] a = doctorDetails[i].split(":");
map.put(a[0], a[1]);
c.setDoctorDetailsMap(map);
}
System.out.println("Enter the doctor id to be searched");
String id = sc.next();
sc.nextLine();
String result = c.findSpecialization(id);
if (result != null) {
if (!result.equalsIgnoreCase("Invalid doctor id")) {
System.out.println("The given doctor " + id + " is specialized in " + result);
} else {
System.out.println("Invalid doctor id");
}
} else {
System.out.println("Invalid doctor id");
}
System.out.println("Enter the specialization to be searched");
String specialization = sc.nextLine();
list1 = c.findDoctorsBasedOnTheGivenSpecialization(specialization);
if (list1 != null && list1.size() >= 1) {
System.out.println("Doctors specialized in " + specialization + " are");
for (String s : list1) {
System.out.println(s);
}
} else {
System.out.println("No doctors were found for the given specialization");
}
}
}