

# Gymnasium Appointment System

---

## What is Gymnasium Appointment System?

Gymnasium Appointment System (GAS) is a software that help a gymnasium center administrative assistant to manage gymnasium information and trainee appointments. The major activities include registering trainees and making appointments. All Therapist, Kinesiologists, Trainees, and Gymnasium need to be registered in the system in order to make and cancel an appointment.

## How Online Gymnasium Appointment System works?

The Gymnasium related information is required to be registered in the system before making appointments. These information include:

- Therapist, Kinesiologists, Trainees.
- Gymnasium details
- Trainees details
- Appointment details

An administrative assistant in a Trainees center can use the system to:

- Make an appointment for a Trainees
- Cancel an appointment for a Trainee
- Print a report for each Trainee

To use the Trainees appointment system, initial parameters have to be set one time. These parameters are the total number of Therapists, total number of Kinesiologists, total number of Trainees, total number of Gymnasium and the number of appointments the Gymnasium system have.

After successful loading of required data, the GAS is geared up to perform various activities. The actual activity starts when an administrative assistant make an appointment for a Trainee. An appointment cannot be booked for more than one Trainee. An administrative assistant can also cancel the appointment.

The system records each and every activity performed including making appointments, and canceling appointment. The system also generates individual reports for each Trainee that has the complete appointments history. You are required to develop the Gymnasium Appointment System simulation software to simulate various Gymnasium Appointment System oriented functionalities.

### **The Initial Procedure of the Program**

Your program will use File I/O to read input from a given file called **input.txt**. To set the initial parameters, your program will first read the following parameters from [input.txt]:

1. total number of Therapist
2. total number of Kinesiologists
3. total number of Trainees
4. total number of Gymnasium
5. total number of appointments

These parameters will be in the first line of the file [input.txt]. To read from the file, the system should first check if the file [input.txt] exists or not and will display an error message if the file does not exist. After reading the parameters, they will be used to create **ArrayList** for:

- Therapist objects
- Kinesiologists
- Trainees objects
- Trainees objects and each Trainee object has an array of appointments

### **Other Objects:**

- Gymnasium objects
- Appointments objects

a) The first number (5) in the file refers to the number of <b>Therapist</b> in the System [ means system will accept ONLY Five Therapist records details]
b) The second number (4) refers to the number of <b>Kinesiologists</b> in the system [ means system will accept ONLY Four Kinesiologists records details]
c) The third number ( 6 ) refers to the number of <b>Trainees</b> in the system [ means system will accept ONLY Six Trainees records details]
d) The fourth number ( 2) refers to the number of <b>Gymnasium</b> in the system [ means system will accept ONLY Two Gymnasium records details]
e) The fifth number ( 24 ) refers to the number of <b>Appointment</b> in the system [ means system will accept ONLY Twenty Four <b>Trainees</b> records details]

For simplicity, the UML class diagrams is provided, and it shows the involved objects and the relationships among them. According to the given UML, you are required to define java classes that describe the properties, behaviors and the relationships of the identified objects. As depicted in UML figure, there are eight classes.

### Details of Classes

You have to create Eight classes in this program.

- Person class is a super class of FitnessExpert and Trainee classes.
- FitnessExpert class is a sub-class of Person class.
- Therapist class is a sub-class of FitnessExpert class.
- Kinesiologist class is a sub-class of FitnessExpert class.
- Trainee class is a sub-class of Person class.
- Gimnasium class.
- Appointment class.
- GimnasiumAppointmentSystem class is the application class to create objects and invoke appropriate methods for program to execute successfully.

## Implementation

The system (your program) will read the following commands to perform one of the program methods. Following is the details of these functionalities:

### SET\_SET\_Therapist

This command will add the details of all the **Therapist** to the system. Each **Therapist** has 10 parameters as shown in below example:

Example
Ahmed_Ali saudi 1980 2 13 M 0555777999 AZahra false Counseling

Following are the details of require parameters:

Sr. No.	Field Name	Data Type	Representing...
1.	<b>name</b>	String	Name of Therapist
2.	<b>nationality</b>	String	Nationality of Therapist
3.	<b>year</b>	int	Year of birth
4.	<b>month</b>	int	Month of birth
5.	<b>day</b>	int	Day of birth
6.	<b>gender</b>	char	Gender of Therapist
7.	<b>phone</b>	string	Phone number of Therapist
8.	<b>address</b>	String	Address of Therapist
9.	<b>onLeave</b>	boolean	If the Therapist is on leave
10.	<b>specialty</b>	String	Specialty of the Therapist

### SET\_Kinesiologist

This command will add the details of all the **Kinesiologists** to the system. Each **Kinesiologist** has 10 parameters as shown in below example:

Example
Layla_Ali egyptian 1980 2 1 F 0555777888 Alfaisaliyah false 5

Following are the details of require parameters:

Sr. No	Field	Data Type	Representing...
.	Name		
1.	<b>name</b>	String	Name of Kinesiologists
2.	<b>nationality</b>	String	Nationality of Kinesiologists
3.	<b>year</b>	int	Year of birth
4.	<b>month</b>	int	Month of birth
5.	<b>day</b>	int	Day of birth
6.	<b>gender</b>	char	Gender of Kinesiologists
7.	<b>phone</b>	string	Phone number of Kinesiologists
8.	<b>address</b>	String	Address of Kinesiologists
9.	<b>onLeave</b>	Boolean	If the Kinesiologists is on leave
10.	<b>experience Years</b>	int	years of experience

### SET\_Trainee

This command will add the details of all the **Trainees** to the system. Each **Trainees** has 9 parameters as shown in below example:

Example
Omar_Anwar saudi 1984 5 13 M 0544477999 AZahra Yoga

Following are the details of require parameters:

Sr. No.	Field Name	Data Type	Representing...
---------	------------	-----------	-----------------

1.	<b>name</b>	String	Name of Trainee
2.	<b>nationality</b>	String	Nationality of Trainee
3.	<b>year</b>	int	Year of birth
4.	<b>month</b>	int	Month of birth
5.	<b>day</b>	int	Day of birth
6.	<b>gender</b>	char	Gender of Trainee
7.	<b>phone</b>	string	Phone number of Trainee
8.	<b>address</b>	String	Address of Trainee
9.	<b>service</b>	String	service of Trainee

### SET\_Gimnasium

This command will add the details of all the gymnasium to the system. Each gymnasium has 2 parameters as shown in below example:

Example	
1	2

Following are the details of require parameters:

Sr. No.	Field Name	Data Type	Representing...
1	<b>gimnasiumNo</b>	int	Gymnasium number
2	<b>Gymnasium_Floor</b>	int	Gymnasium floor

### SET\_Appointment

This command will add the details of all the appointment slots to the system. The command has 6 parameters as shown in below example:

Example					
1	2	2020	3	12	10

Following are the details of require parameters:

Sr. No.	Field Name	Data Type	Representing...
1.	<b>gymnasiumNo</b>	int	Gymnasium number
2.	<b>gymnasiumFloor</b>	int	Gymnasium Floor
3.	<b>year</b>	int	Year of the appointment
4.	<b>month</b>	int	Month of the appointment
5.	<b>day</b>	int	Day of the appointment
6.	<b>startTime</b>	int	Start time of the appointment

### ReserveAnAppointment

This command will Reserve an appointment for the Trainee based on the entry data as shown in below example:

#### Input Example

```
RSRVAppointment Omar_Anwar 1 2 2020 3 12 11
```

Following are the details of require parameters:

Sr. No.	Field Name	Data Type	Representing...
1.	<b>traineeName</b>	String	Name of Trainee
2.	<b>GymnasiumNo</b>	int	Gymnasium number
3.	<b>GymnasiumFloor</b>	int	Gymnasium Floor
4.	<b>year</b>	int	Year of the appointment
5.	<b>month</b>	int	Month of the appointment
6.	<b>day</b>	int	Day of the appointment
7.	<b>startTime</b>	int	Start time of the appointment

Output Example
<p>COMMAND: RESERVE APPOINTMENT</p> <p>Appointment for Trainee Omar_Anwar is done</p> <p>-----</p>

Note
<p>The system should NOT accept to add the appointment if the provided Trainee name doesn't exists in the system. In this case the system will display an appropriate error message like:</p> <p>Trainee Omar_Ali is not registered</p>
<p>If the searched date and time of the appointment is not available in the system, the system will display an appropriate message like:</p> <p>Appointment not found in the Gimnasium appointment list</p>
<p>If the searched date and time of the appointment is taken by other Trainee, the system will display an appropriate message like:</p> <p>Appointment is taken by other Trainee</p>
<p>A Trainee has to have a maximum of 2 appointments registered in the system, if an administrative assistant try to make a third appointment, the system will display an appropriate message like:</p> <p>Appointment for Trainee Omar_Anwar can not be added as he has 2 appointment registered</p>

### cancelAppointment

This command will cancel a particular appointment of a Trainee based on the provided entry data as shown in below example:

Input Example
---------------



```
CancelAppointment Adel_Mustafa 1 2 2020 3 12 17
```

Following are the details of require parameters:

Sr. No.	Field Name	Data Type	Representing...
1.	<b>TraineeName</b>	String	Name pf Trainee
2.	<b>Gymnasium No</b>	int	Gymnasium number
3.	<b>GymnasiumFloor</b>	int	Gymnasium Floor
4.	<b>year</b>	int	Year of the appointment
5.	<b>month</b>	int	Month of the appointment
6.	<b>day</b>	int	Day of the appointment
7.	<b>startTime</b>	int	Start time of the appointment

#### Output Example

```
COMMAND: CANCEL APPOINTMENT
```

```
Appointment for Trainee Adel_Mustafa is Canceled  
Appointment is Canceled from the Gymnasium appointment list and  
available for other Trainee.
```

#### Note

The system should NOT accept to cancel the appointment if the provided Trainee name doesn't exists in the system. In this case the system will display an appropriate error message like:

```
Trainee: Adala_Hassan is not registered
```

The system will not cancel the appointment if the appointment is not registered in the system for this Trainee. In this case the system will display an appropriate message like:

Appointment not found in Trainee Adel\_Mustafa records

If an appointment is canceled for the given Trainee, this Appointment should be setting as available in the system for other Trainees.

### PrintTraineeRecord

This command will print Trainee APPOINTMENTS record for a specific Trainee. The command has 1 parameter as shown in below example:

#### Input Example

```
PrintTraineeRecord Omar_Anwar
```

Following are the details of require parameters:

Sr. No.	Field Name	Data Type	Representing...	Example
1	Trainee Name	String	Name of the Trainee	Sara Ayman

#### Output Example

```
COMMAND: PRINT Trainee APPOINTMENTS RECORD
```

```
Appointment for Omar Anwar are:
```

```
Appointment number: 1 Gimnasium: Number : 1 Floor: 2 Date: 12/3/2023 Time: 11:00
```

```
Appointment number: 2 Gimnasium: Number : 1 Floor: 2 Date: 13/3/2023 Time: 19:00
```

-----

Note
<p>The system should NOT accept to print the record if the provided Trainee name doesn't exists in the system. In this case the system will display an appropriate error message like:</p> <p>Trainee <b>Ali_Hassan</b> is not registered</p>
<p>The system should NOT accept to print the record if the Trainee doesn't have any registered appointment. In this case the system will display an appropriate error message like:</p> <p>Appointment for <b>Adel_Mustafa</b> are: Trainee <b>Adel_Mustafa</b> has no appointent</p>

### Important Notes:

- Your program output must be exactly same as given sample output files.
- Your display should be in a readable form.
- Organize your code in separated methods.
- Document your code with comments.
- Use meaningful variables.
- Use dash lines between each method.
- Delayed submission will not be accepted and there will not be any extension of the project.

### Deliverables:

You should submit one zip file containing all java codes: BA1587412P2.java where BA is your section, 1587412 your ID and p2 is program 2.

NOTE: your name, ID, and section number should be included as comments in all files!

### Input and Output Format

Your program must generate output file in a similar format to the sample run provided.

