

# Clinic Appointment System

---

## Assignment 3

**CPCS203 Programming-II**

## What is Clinic Appointment System?

Clinic Appointment System (CAS) is a software that help a medical center administrative assistant to manage clinical information and patient appointments. The major activities include registering patients and making appointments. All physicians, nurses, patients, and clinics need to be registered in the system in order to make, and cancel an appointment.

## How Online Clinic Appointment System works?

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

- Healthcare practitioners: physicians and nurses.
- Clinic details
- Patients details

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

- Add physicians details in the system
- Add nurses details in the system
- Add clinics details in the system
- Add patients details in the system
- Make an appointment for a patient
- Cancel an appointment for a patient
- Print a report for each patients

To use the clinic appointment system, an administrative assistant has to set **initial parameters** one time. These parameters are the **total number of clinics** the system has to manage, the **total number of patients** the system has to create records for, the **total number of physicians and nurses in a clinic center** and the **number of appointments allowed** the system has to make.

After successful loading of required data, the CAS is geared up to perform various activities. The actual activity starts when an administrative assistant make an appointment for a patient. An appointment cannot be booked for more than one patient. 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 patient that has all the patient's details including his/her complete appointments history. You are required to develop the Clinic Appointment System simulation software to simulate various Clinic 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 clinics**
2. **total number of patients**
3. **total number of physicians**
4. **total number of nurses**
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.

a)	The first number (5) in the file refers to the number of <i>Physician</i> in the System [ means system will accept ONLY TEN physician records details]
b)	The second number (4) refers to the number of <i>Nurse</i> in the system [ means system will accept ONLY FIVE Nurse records details]
c)	The sixth number ( 6 ) refers to the number of <i>Patient</i> in the system [ means system will accept ONLY Fifteen Patients records details]
d)	The third number ( 2 ) refers to the number of <i>Clinic</i> in the system [ means system will accept ONLY TEN clinic records details]
e)	The sixth number ( 24 ) refers to the number of <i>Appointment</i> in the system [ means system will accept ONLY Fifteen Patients 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. Five of them to model objects. Four for modeling physical objects such as physician, nurse, patients and clinic, while the fifth for modelling non- physical objects such as appointment. Person and healthcarePractitioner classes are general ones used to avoid redundancy in physician, nurse and patient classes. ClinicAppointmentSystem class represents the high abstraction level of the system, and all functions (methods) of the system invoked through the object of this class.

Please note you are not forced to shape your classes and their relationships as exactly as the ones in the given UML. However, you have to apply the inheritance concept to avoid redundancy and also do not change the skeleton of clinicAppointmentSystem class, because it represents the high abstraction functionalities of the system.

## Details of Classes

You have to create **Eight classes** in this program.

- **Person** class is super class of ALL the classes of this assignment.
- **HealthcarePractitioner** class is a sub-class of Person.
- **Physician** class is a sub-class of HealthcarePractitioner.
- **Nurse** class is a sub-class of HealthcarePractitioner.
- **Patient** class is a sub-class of Person.
- **Clinic** class to store Clinics details.
- **Appointment** class to store Appointments details.
- **ClinicAppointmentSystem** to create objects and invoke appropriate methods for program to execute successfully.

## Implementation

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

### addPhysician

This command will add a new Physician to the system. The command has 10 parameters as shown in below example:

Example
<pre>addPhysician("Hamed_Hamadan", "Egyptian",1978, 04, 04, 'M', 505599882, Jeddah, False, "pediatrician")</pre>

Following are the details of require parameters:

Sr. No.	Field Name	Data Type	Representing...	Example
1.	name	String	Name of physician	Hamed_Hamadan
2.	nationality	String	Nationality of physician	Egyptian
3.	year	int	Year of birth	1978
4.	month	int	Month of birth	04
5.	day	int	Day of birth	04
6.	gender	char	Gender of physician	M
7.	phone	int	Phone number of physician	36640454
8.	address	String	Address of physicians	Riyadh
9.	onLeave	Boolean	If the physician is on leave	False
10.	specialty	String	Specialty of the physician	pediatrician

### addNurse

This command will add a new Nurse to the system. The command has 10 parameters as shown in below example:

Example
<pre>addNurse("Hala Taher", "Saudi",1968, 10, 24, 'F', 505588882, Jeddah, False, 2)</pre>

Following are the details of require parameters:

Sr. No.	Field Name	Data Type	Representing...	Example
1.	name	String	Name of nurse	Tala Taher
2.	nationality	String	Nationality of nurse	Saudi
3.	year	int	Year of birth	1969
4.	month	int	Month of birth	10

5.	<b>day</b>	int	Day of birth	24
6.	<b>gender</b>	char	Gender of nurse	F
7.	<b>phone</b>	int	Phone number of nurse	505588882
8.	<b>address</b>	String	Address of nurse	Jeddah
9.	<b>onLeave</b>	Boolean	If the nurse is on leave	False
10.	<b>experienceYears</b>	int	Number years of experience	2

### addPatient

This command will add a new Patient to the system. The command has 9 parameters as shown in below example:

Example				
<pre>addPatient("Alaa_Ahmad_Zahrani","Saudi",1973,10,1 'F',534528754,"Jeddah","Diabetes ")</pre>				

Following are the details of require parameters:

Sr. No.	Field Name	Data Type	Representing...	Example
1.	<b>name</b>	String	Name of nurse	Alaa Zahrani
2.	<b>nationality</b>	String	Nationality of nurse	Saudi
3.	<b>year</b>	int	Year of birth	1973
4.	<b>month</b>	int	Month of birth	10
5.	<b>day</b>	int	Day of birth	1
6.	<b>gender</b>	char	Gender of nurse	F
7.	<b>phone</b>	int	Phone number of nurse	534528754
8.	<b>address</b>	String	Address of nurse	Jeddah
9.	<b>diagnosis</b>	String	Diagnosis of patient	Diabetes

### addClinic

This command will add a new Clinic to the system. The command has 2 parameters as shown in below example:

Example				
<pre>addClinic(501,'F')</pre>				

Following are the details of require parameters:

Sr. No.	Field Name	Data Type	Representing...	Example
1	<b>clinicNo</b>	int	Clinic number	501
2	<b>floor</b>	char	Clinic floor	F

### addAppointment

This command will add an appointment slot to the system. The command has 7 parameters as shown in below example:

Example
<code>addAppointment(103,'F', 2020, 04, 01, 13, True)</code>

Following are the details of require parameters:

Sr. No.	Field Name	Data Type	Representing...	Example
1.	<b>clinicNo</b>	int	Clinic number	103
2.	<b>clinicFloor</b>	char	Clinic Floor	F
3.	<b>year</b>	int	Year of the appointment	2020
4.	<b>month</b>	int	Month of the appointment	04
5.	<b>day</b>	int	Day of the appointment	01
6.	<b>startTime</b>	int	Start time of the appointment	13
7.	<b>available</b>	Boolean	If the appointment is available	True

#### Note

The system should NOT accept to add an appointment slot with end time or start time more than 20. In this case the system will display an appropriate message like:

**The end time and the start time should be less than 20**

### makeAnAppointment

This command will add an appointment for the patient based on the entry data of clinicNo, year, month, day, and time of the appointments. The command has 7 parameters as shown in below example:

Example
<code>makeAnAppointment ("Alaa Mohammed" , 103 , F, 2020, 04, 01, 13)</code>

Following are the details of require parameters:

Sr. No.	Field Name	Data Type	Representing...	Example
1.	<b>patientName</b>	String	Name pf patient	"Alaa Mohammed"
2.	<b>clinicNo</b>	int	Clinic number	103
3.	<b>clinicFloor</b>	char	Clinic Floor	F
4.	<b>year</b>	int	Year of the appointment	2020
5.	<b>month</b>	int	Month of the appointment	04
6.	<b>day</b>	int	Day of the appointment	01
7.	<b>startTime</b>	int	Start time of the appointment	13

#### Note

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

**There is no patient with name "Amani Kalid"**

If the searched date and time of the appointment is not available in the system, the system will display an appropriate message like:

**The appointment is not registered in the system.**

If the searched date and time of the appointment is taken by other patient, the system will display an appropriate message like:

**This appointment is taken**

A patient 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:

**This patient has reached the maximum number of registered appointments**

#### cancelAppointment

This command will cancel a particular appointment of a patient based on the provided entry data of clinicNo, year, month, day, and time of the appointments. The command has



7 parameters as shown in below example:

Example	
<b>cancelAppointment</b>	("Alaa Mohammed" , 103 , F, 2020, 04, 01, 13)

Following are the details of require parameters:

Sr. No.	Field Name	Data Type	Representing...	Example
1.	<b>patientName</b>	String	Name pf patient	"Alaa Mohammed"
2.	<b>clinicNo</b>	int	Clinic number	103
3.	<b>clinicFloor</b>	char	Clinic Floor	F
4.	<b>year</b>	int	Year of the appointment	2020
5.	<b>month</b>	int	Month of the appointment	04
6.	<b>day</b>	int	Day of the appointment	01
7.	<b>startTime</b>	int	Start time of the appointment	13

Note
The system should NOT accept to add the appointment if the provided patient name doesn't exists in the system. In this case the system will display an appropriate error message like:  <b>There is no patient with name "Amani Kalid"</b>
The system will not cancel the appointment if the appointment is not registered in the system for this patient. In this case the system will display an appropriate message like:  <b>There is not registered appointment with these details for this patient</b>
If an appointment is canceled for the given patient, it should be available in the system for other patients

### **printPatientRecord**

This command will print a report for a specific patient. The command has 1 parameter as shown in below example:

#### Example

```
printPatient (" Sara Ayman")
```

Following are the details of require parameters:

Sr. No.	Field Name	Data Type	Representing...	Example
1	<b>patientName</b>	String	Name of the patient	Sara Ayman

#### Note

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

```
There is no patient with name "Amani Kalid"
```

The system should NOT accept to print the record if the patient doesn't have a registered appointment. In this case the system will display an appropriate error message like:

```
This patient does not have an appointment
```

### Important Notes:

- Use of class & object, arrays of Object, passing object to method and Inheritance is mandatory.
- Use of Files, Reading/Writing from/on files and String , StringBuilder methods.
- 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: BA1587412P3\_OCAS.java where BA is your section, 1587412 your ID and p3 is program 3.
- **NOTE: your name, ID, and section number should be included as comments in all files!**