REPORT

Adv OOP Lab

001910501005

Atanu Ghosh

JU BCSE UG-II Even Sem

Java Assignment - 1

Q1: Design a STUDENT class to store roll, name, course, admission date and marks in 5 subjects taken from user. Create an array of STUDENT objects. Provide methods corresponding to admission date and receiving marks, preparing mark sheet. Support must be there to show the number of students who have taken admission. Inherit Student class and override the input method so as to input the department of each student. Search and display a sorted list of students of one department or students based on scoring criteria. Create an arraylist of students and remove a student based on certain criterion and then call gc() and check for free memory.

NOTE: Student roll/ID should be fixed length which include department name, admission year and roll and it should be auto generated (no random roll no will be accepted). Consider at least four departments. 1st four characters: dept, 2nd two characters : year, last three characters: roll no

Example: if a student is in dept. Of Computer Science and Engg.

1st admission : BCSE18001 (considering max 100 students in each dept.)

2nd admission : BCSE1 8002

If a student is in dept. Of Electronics and Telecommunication Engg.

1st admission : ETCE18001 (considering max 100 students in each dept.)

3rd admission : ETCE18003

UML Dagram :

Output :

```
atanu@dell:~/Documents/OOP Adv Lab/java_assignments1/q1$ java Q1

Main Menu
-------------
1. Add Student
2. Display Marksheet of a Student by Roll
3. Sort students of a Department by marks
4. Remove a Student
5. View Total no. of students
6. Call GarbageCollector
7. Show Free Memory
-------------
Enter your choice: 1

Enter name: Atanu Ghosh
Enter course: BCSE
Enter admission date (DD-MM-YYYY): 01-02-2019
Enter department name: CSE
Roll number allotted:- BCSE19001
Enter marks in 5 subjects: 98 87 76 88 79
Student added successfully.


Do you want to continue (y/n) ? y

Main Menu
-------------
1. Add Student
2. Display Marksheet of a Student by Roll
3. Sort students of a Department by marks
4. Remove a Student
5. View Total no. of students
6. Call GarbageCollector
7. Show Free Memory
-------------
Enter your choice: 1

Enter name: Sourish Pal
Enter course: BETCE
Enter admission date (DD-MM-YYYY): 03-04-2019
Enter department name: ETCE
Roll number allotted:- ETCE19001
Enter marks in 5 subjects: 12 23 34 45 56
Student added successfully.


Do you want to continue (y/n) ? y

Main Menu
-------------
1. Add Student
```

```
Enter your choice: 2

Enter roll number: ETCE19001

Student found.

Roll: ETCE19001         Name: Sourish Pal
Course: BETCE
Admission Date: 2019-04-03
Marks in 5 subjects:
12.0 23.0 34.0 45.0 56.0
Total marks = 170.0     Percentage = 34.0

Do you want to continue (y/n) ? y

Main Menu
-------------
1. Add Student
2. Display Marksheet of a Student by Roll
3. Sort students of a Department by marks
4. Remove a Student
5. View Total no. of students
6. Call GarbageCollector
7. Show Free Memory
-------------
Enter your choice: 3

Enter department code: ETCE19001
Department does not exist.


Do you want to continue (y/n) ? y

Main Menu
-------------
1. Add Student
2. Display Marksheet of a Student by Roll
3. Sort students of a Department by marks
4. Remove a Student
5. View Total no. of students
6. Call GarbageCollector
7. Show Free Memory
-------------
Enter your choice: 4

Enter roll number: ETCE19001

Student with roll number ETCE19001 removed successfully.


Do you want to continue (y/n) ?
```

```
Main Menu
--------------
1. Add Student
2. Display Marksheet of a Student by Roll
3. Sort students of a Department by marks
4. Remove a Student
5. View Total no. of students
6. Call GarbageCollector
7. Show Free Memory
--------------
Enter your choice: 5

No of students registered: 1


Do you want to continue (y/n) ? y

Main Menu
--------------
1. Add Student
2. Display Marksheet of a Student by Roll
3. Sort students of a Department by marks
4. Remove a Student
5. View Total no. of students
6. Call GarbageCollector
7. Show Free Memory
--------------
Enter your choice: 6

Garbage Collector called.


Do you want to continue (y/n) ? y

Main Menu
--------------
1. Add Student
2. Display Marksheet of a Student by Roll
3. Sort students of a Department by marks
4. Remove a Student
5. View Total no. of students
6. Call GarbageCollector
7. Show Free Memory
--------------
Enter your choice: 7

Free Memory:- 7230624


Do you want to continue (y/n) ?
```

Q2 : Design a system for the following scenario:

1. An item list contains item code, name, rate, and quantity for several items.

2. Whenever a new item is added in the list uniqueness of item code is to be checked. Register a new product with its price.

3. Time to time rate of the items may change.

4. Whenever an item is issued or received existence of the item is checked and quantity is updated.

5. In case of issue, availability of quantity is also to be checked.

6. User may also like to know price/quantity available for an item.

7. Find how many items cost more than a given amount. The amount will be a parameter.

8. Remember that the methods have to return an error code if for example an invalid item code is given

NOTE:

● The system should be maintained by two types of user, one is Stock entry operator(SEO) and other is Shopkeeper (SK) and SEO will be the first operator in default case.

● The SEO primarily maintain first 3 operations but SEO users can also maintain all operations (1 to 8)

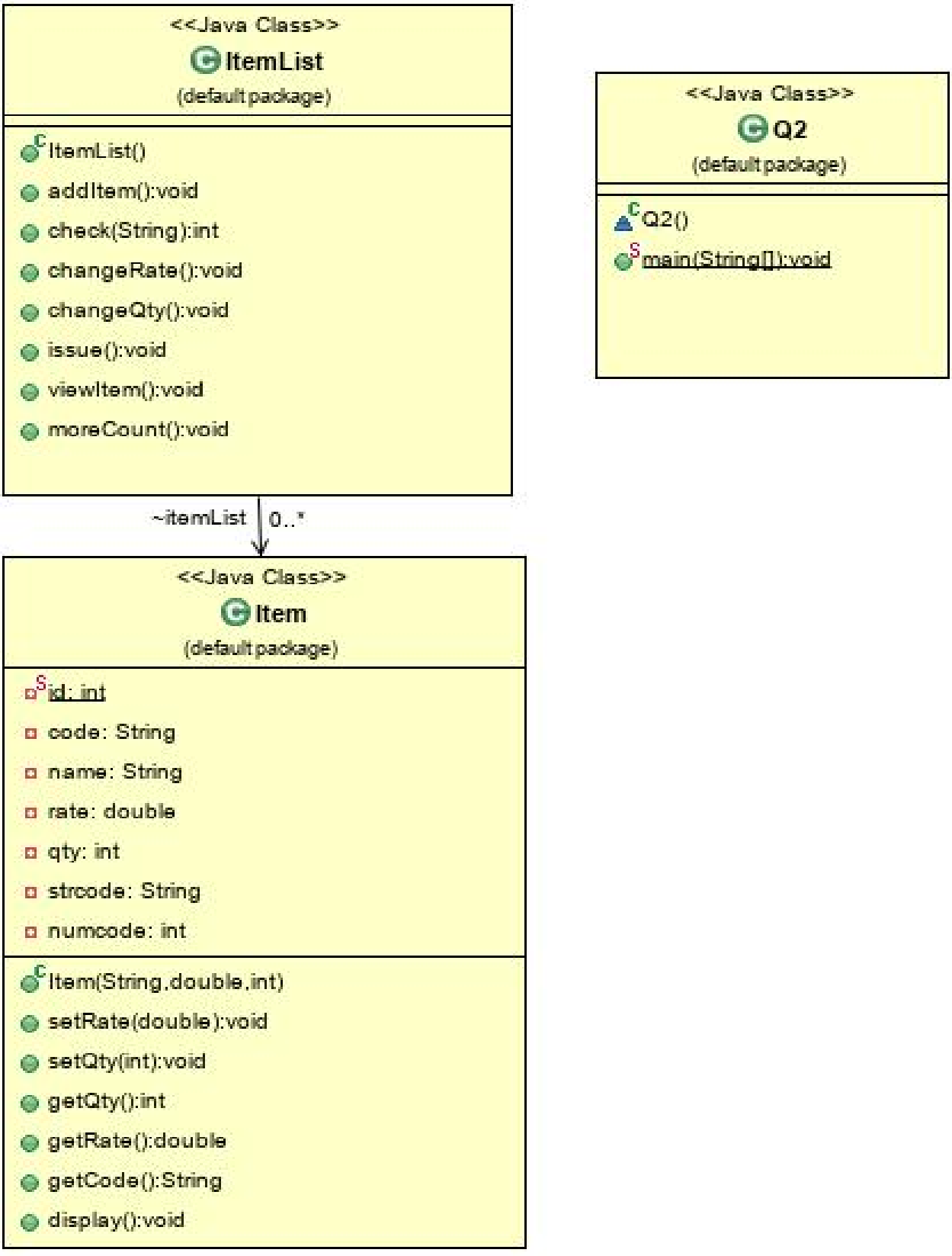
● SK users can only operates on 4 to 8.

● System should be used for a specific shop type. Ex. Electronics, Book, Grocer etc.. You can design your system for any one.

● Item Code should be auto generated that includes item name and entry order(1,2,3...) Example: for Electronics shop Item name entry order Item Code Laptop 3 LAP003 Mobile 2 MOB002 Monitor 10 MON010 Mouse 1 MOU001

Example: for Electronics shop Item name entry order Item Code Laptop 3 LAP003 Mobile 2 MOB002 Monitor 10 MON010 Mouse 1 MOU001

UML Diagram :


<<Java Class>>
ItemList
(default package)
ItemList()
addItem():void
check(String):int
changeRate():void
changeQty():void
issue():void
viewItem():void
moreCount():void
~itemList
0..*
<<Java Class>>
Item
(default package)
id: int
code: String
name: String
rate: double
qty: int
strcode: String
numcode: int
Item(String,double,int)
setRate(double):void
setQty(int):void
getQty():int
getRate():double
getCode():String
display():void
<<Java Class>>
Q2
(default package)
Q2()
main(String[]):void

Output :

```
atanu@dell:~/Documents/OOP Adv Lab/java_assignments1/q2$ javac Q2.java
atanu@dell:~/Documents/OOP Adv Lab/java_assignments1/q2$ java Q2
1. SEO
2. Shopkeeper
3.Exit
Enter choice :1
1. Add new item
2. Change rate of existing item
3. Update quantity
4. Issue an item
5. View price or quantity of an item
6. Find how many items cost more than a given amount
7. Go back
Enter choice :1
Enter name of item
Laptop
Enter rate
50000
Enter qty
10
Code :LAP010
Name: Laptop
Rate: 50000.0
Quantity: 10
1. Add new item
2. Change rate of existing item
3. Update quantity
4. Issue an item
5. View price or quantity of an item
6. Find how many items cost more than a given amount
7. Go back
Enter choice :1
Enter name of item
Mouse
Enter rate
2000
Enter qty
10
Code :MOU010
Name: Mouse
Rate: 2000.0
Quantity: 10
1. Add new item
2. Change rate of existing item
3. Update quantity
4. Issue an item
5. View price or quantity of an item
6. Find how many items cost more than a given amount
7. Go back
Enter choice :2
Enter item code
```

```
Enter choice :2
Enter item code
LAP010
Enter new rate
60000
1. Add new item
2. Change rate of existing item
3. Update quantity
4. Issue an item
5. View price or quantity of an item
6. Find how many items cost more than a given amount
7. Go back
Enter choice :3
Enter item code
LAP010
Enter new quantity
20
1. Add new item
2. Change rate of existing item
3. Update quantity
4. Issue an item
5. View price or quantity of an item
6. Find how many items cost more than a given amount
7. Go back
Enter choice :5
Enter item code
LAP010
Code :LAP010
Name: Laptop
Rate: 60000.0
Quantity: 20
1. Add new item
2. Change rate of existing item
3. Update quantity
4. Issue an item
5. View price or quantity of an item
6. Find how many items cost more than a given amount
7. Go back
Enter choice :4
Enter item code
LAP010
Enter quantity
34
Insufficient quantity
1. Add new item
2. Change rate of existing item
3. Update quantity
4. Issue an item
5. View price or quantity of an item
6. Find how many items cost more than a given amount
7. Go back
```

```
Enter choice :4
Enter item code
LAP010
Enter quantity
11
Rate: 60000.0
Quantity: 11
Total cost: 660000.0
1. Add new item
2. Change rate of existing item
3. Update quantity
4. Issue an item
5. View price or quantity of an item
6. Find how many items cost more than a given amount
7. Go back
Enter choice :6
Enter amt
13
No of items above 13.0 is 2
1. Add new item
2. Change rate of existing item
3. Update quantity
4. Issue an item
5. View price or quantity of an item
6. Find how many items cost more than a given amount
7. Go back
Enter choice :7
Invalid choice
1. SEO
2. Shopkeeper
3.Exit
Enter choice :2
1. Update quantity
2. Issue an item
3. View price or quantity of an item
4. Find how many items cost more than a given amount
5. Go back
Enter choice :1
Enter item code
LAP010
Enter new quantity
40
1. Update quantity
2. Issue an item
3. View price or quantity of an item
4. Find how many items cost more than a given amount
5. Go back
Enter choice :2
Enter item code
LAP010
Enter quantity
```

```
Enter choice :2
Enter item code
LAP010
Enter quantity
45
Insufficient quantity
1. Update quantity
2. Issue an item
3. View price or quantity of an item
4. Find how many items cost more than a given amount
5. Go back
Enter choice :2
Enter item code
LAP010
Enter quantity
33
Rate: 60000.0
Quantity: 33
Total cost: 1980000.0
1. Update quantity
2. Issue an item
3. View price or quantity of an item
4. Find how many items cost more than a given amount
5. Go back
Enter choice :3
Enter item code
LAP010
Code :LAP010
Name: Laptop
Rate: 60000.0
Quantity: 7
1. Update quantity
2. Issue an item
3. View price or quantity of an item
4. Find how many items cost more than a given amount
5. Go back
Enter choice :4
Enter amt
15
No of items above 15.0 is 2
1. Update quantity
2. Issue an item
3. View price or quantity of an item
4. Find how many items cost more than a given amount
5. Go back
Enter choice :5
Invalid choice
1. SEO
2. Shopkeeper
3.Exit
Enter choice :
```

Q3 : Write a program Parentheses.java that reads in a text stream from standard input and uses a stack to determine whether or not its parentheses are properly balanced. For example, your program should print true for [()]{}{[()()]()} and false for [(]). You need to implement the stack class by yourself

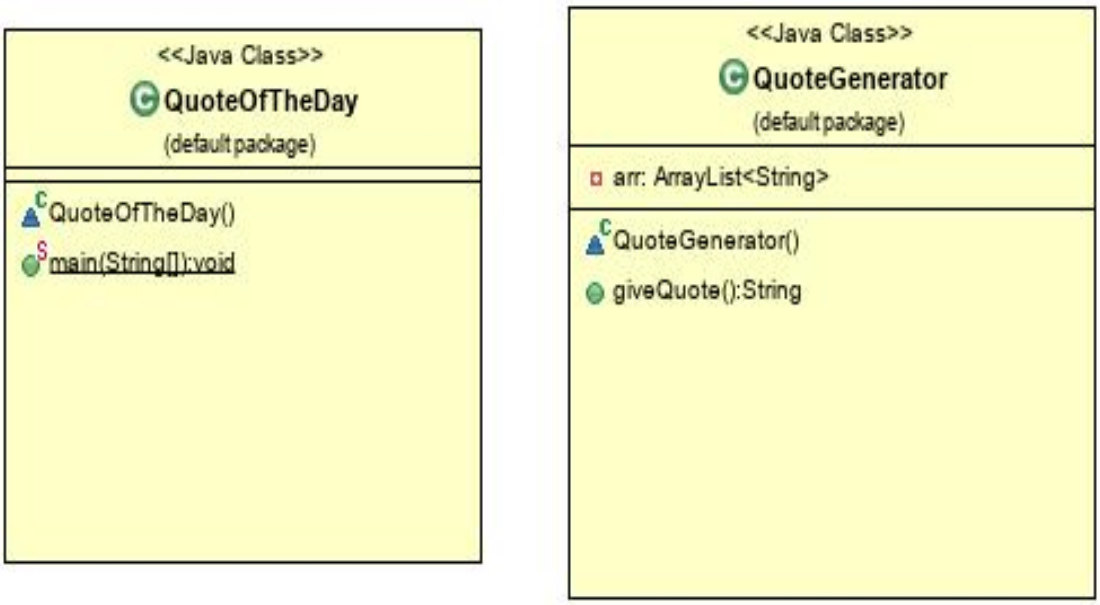
UML Diagram :

Output :

```
atanu@dell:~/Documents/OOP Adv Lab/java_assignments1/q3$ javac Parentheses.java
atanu@dell:~/Documents/OOP Adv Lab/java_assignments1/q3$ java Parentheses
Enter stream of text to check :
{[({})]}
match

press 1 to continue, 0 to exit
1
Enter stream of text to check :
{[)[]}
not match

press 1 to continue, 0 to exit
0
atanu@dell:~/Documents/OOP Adv Lab/java_assignments1/q3$
```

Q4 : Create a class diagram and Java code for the following system and scenario, taking into account the possibility of future extensions. "The system is a command line utility that prints a short 'quote of the day' on the user's terminal when run. To begin with the quote is selected randomly from a set of hard-coded strings within the program itself, but that might change later on -- the quotes might be based on the user's history, the time of day, the date, etc.. Scenario 1. User types "java QuoteOfTheDay" on the command line. 2. System prints out a quote of the day, with an attribution

UML Diagram :



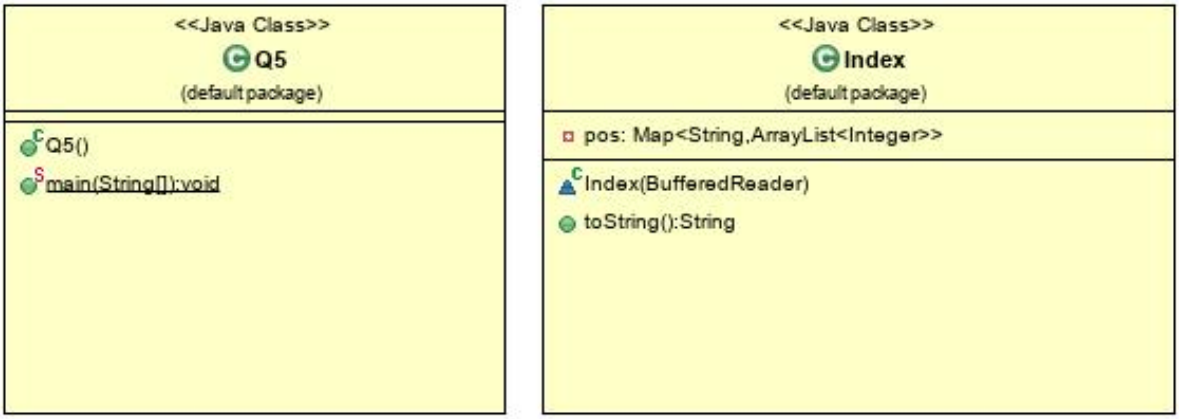
Output :

```
atanu@dell:~/Documents/OOP Adv Lab/java_assignments1/q4$ javac QuoteOfTheDay.java
atanu@dell:~/Documents/OOP Adv Lab/java_assignments1/q4$ java QuoteOfTheDay
Printing current date & time...  Fri May 07 20:42:17 IST 2021
Quote : Be yourself; everyone else is already taken.— Oscar Wilde
atanu@dell:~/Documents/OOP Adv Lab/java_assignments1/q4$ java QuoteOfTheDay
Printing current date & time...  Fri May 07 20:42:29 IST 2021
Quote : Honesty is the first chapter of the book of wisdom. --Thomas Jefferson
atanu@dell:~/Documents/OOP Adv Lab/java_assignments1/q4$ java QuoteOfTheDay
Printing current date & time...  Fri May 07 20:42:30 IST 2021
Quote : God helps them that help themselves. -- Benjamin Franklin
atanu@dell:~/Documents/OOP Adv Lab/java_assignments1/q4$ java QuoteOfTheDay
Printing current date & time...  Fri May 07 20:42:31 IST 2021
Quote : To live is the rarest thing in the world. Most people exist, that is all.— Oscar Wilde
atanu@dell:~/Documents/OOP Adv Lab/java_assignments1/q4$ java QuoteOfTheDay
Printing current date & time...  Fri May 07 20:42:31 IST 2021
Quote : To live is the rarest thing in the world. Most people exist, that is all.— Oscar Wilde
atanu@dell:~/Documents/OOP Adv Lab/java_assignments1/q4$ java QuoteOfTheDay
Printing current date & time...  Fri May 07 20:42:32 IST 2021
Quote : God helps them that help themselves. -- Benjamin Franklin
atanu@dell:~/Documents/OOP Adv Lab/java_assignments1/q4$
```

Q5 : Indexing a book. Write a program that reads in a text file from standard input and compiles an alphabetical index of which words/phrases appear on which lines, as in the following input. Ignore case and punctuation. For each word maintain a list of location on which it appears. Try to use HashTable and/or HashMap class (of java.util).

Note: key of HashMap : word or phrase Value of HashMap can be single or multiple(if multiple time occurs)--

UML Diagram :



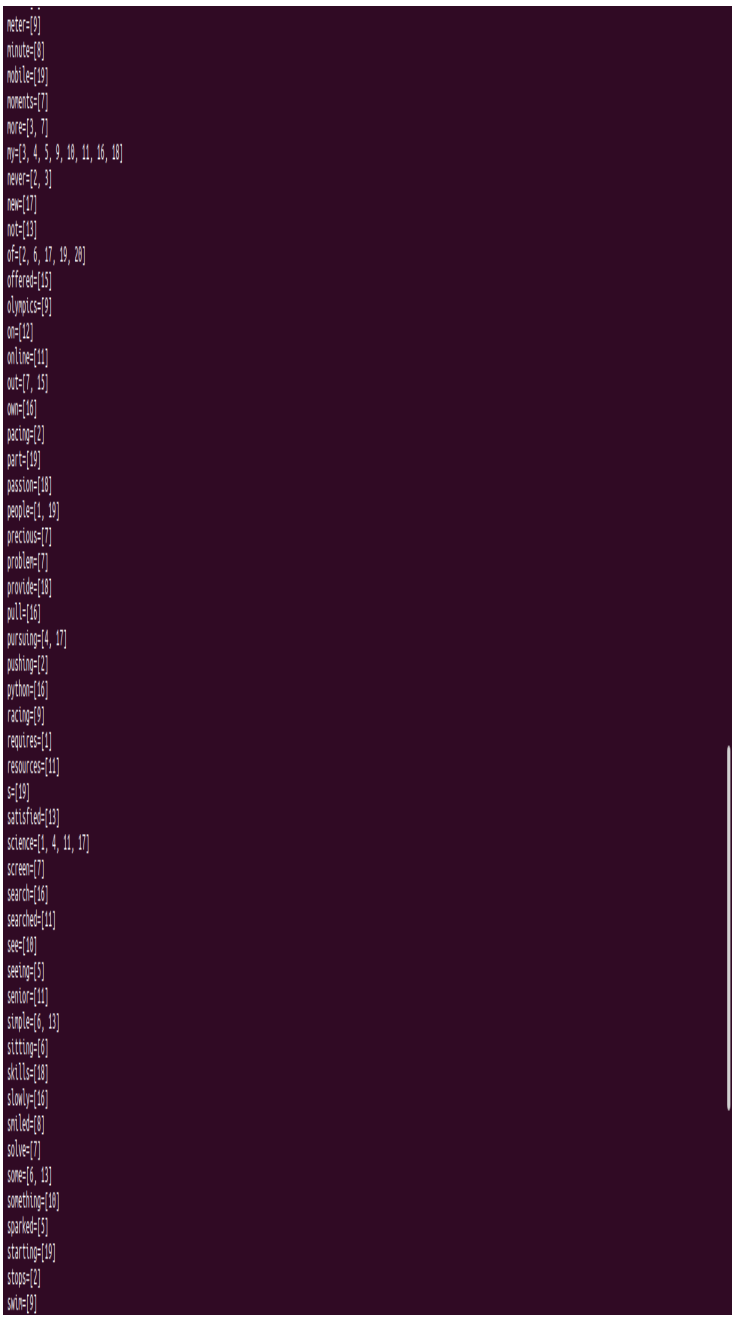
ReadThis.txt :

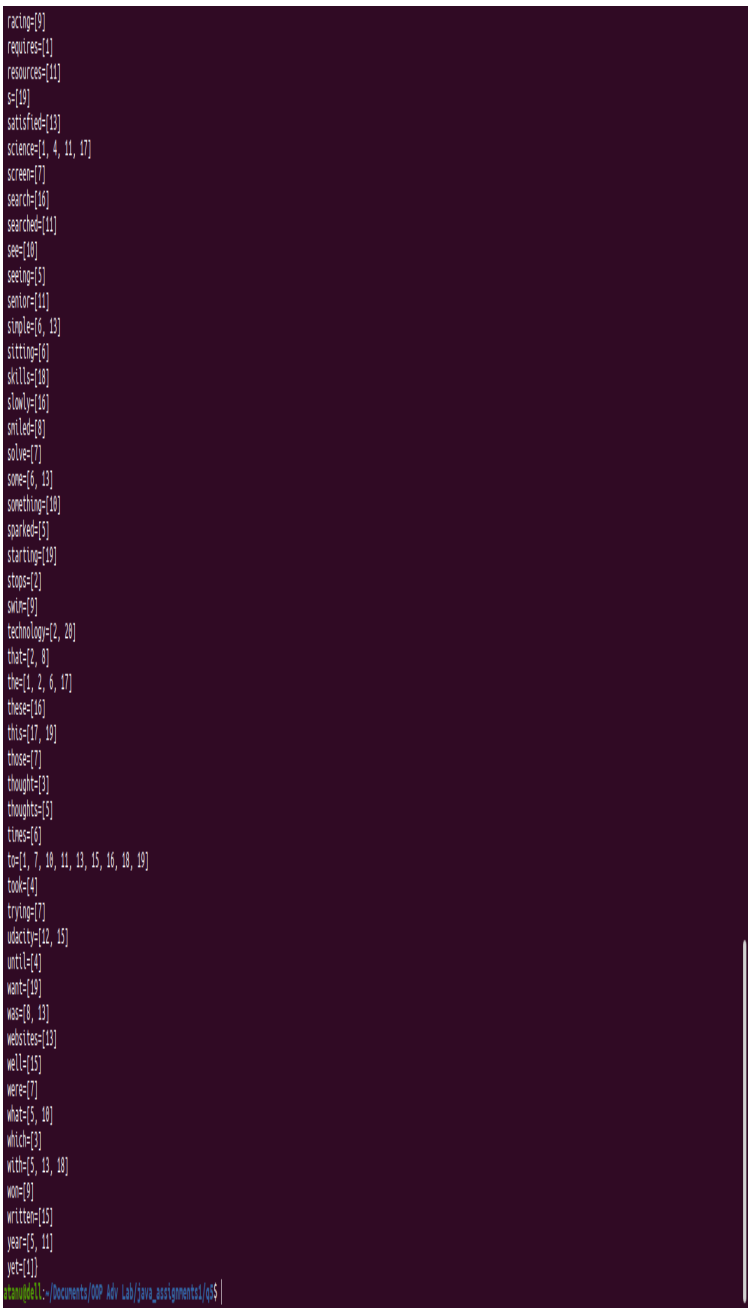
```
Computer science is challenging, and yet dynamic. It requires people in the field to be
keep learning and pushing the limit. That fast-pacing innovation of technology never stops
amazing me, which excites my innate curiosity even more. I, however, had never thought
about pursuing computer science as a career until I took an introductory class in my junior
 year. It sparked my interest and changed my thoughts after seeing what I could do with
some simple line of codes. Despite of many countless times sitting in front of the computer
 screen trying out figure how to do solve a problem, those "Aha" moments were more precious
 and exciting for me. For every minute like that, I was exhilarating and smiled as if I
have just won an Olympics' gold medal in a 100-meter racing swim lap. It fed my hunger for
exploration to see what else I could do, may be something beyond my imagination. Computer
science gradually became my hobby in my senior year. I searched many online resources to
finally find Codeacademy and later Udacity. I first learned about HTML and CSS on
CodeAcademy to make some simple websites, but I was not satisfied with it.

 Meanwhile, I found out about Udacity, offered a well-written and easy-to-follow
introductory course in Python to make my own Google search. These experiences slowly pull
me into the course of a new career, computer science. Pursuing a degree in this field can
provide me knowledge and skills and along with my passion to bring changes to
people's daily lives starting from mobile application. I want to be a part of this
evolution of technology.
```

Output :

```
atanu@dell:~/Documents/OOP Adv Lab/java_assignments1/q5$ javac Q5.java
atanu@dell:~/Documents/OOP Adv Lab/java_assignments1/q5$ java Q5
Enter file path :ReadThis.txt
{=[5, 7, 8, 14, 15]
 100=[9]
 a=[4, 7, 9, 15, 17, 19]
 about=[4, 12, 15]
 after=[5]
 aha=[7]
 along=[18]
 amazing=[3]
 an=[4, 9]
 and=[1, 2, 5, 8, 12, 15, 18]
 application=[19]
 as=[4, 8]
 be=[1, 10, 19]
 became=[11]
 beyond=[10]
 bring=[18]
 but=[13]
 can=[17]
 career=[4, 17]
 challenging=[1]
 changed=[5]
 changes=[18]
 class=[4]
 codeacademy=[12, 13]
 codes=[6]
 computer=[1, 4, 6, 10, 17]
 could=[5, 10]
 countless=[6]
 course=[16, 17]
 css=[12]
 curiosity=[3]
 daily=[19]
 degree=[17]
 despite=[6]
 do=[5, 7, 10]
 dynamic=[1]
 easy=[15]
 else=[10]
 even=[3]
 every=[8]
 evolution=[20]
 excites=[3]
 exciting=[8]
 exhilarating=[8]
 experiences=[16]
 exploration=[10]
 fast=[2]
 fed=[9]
```

```
 fed=[9]
 field=[1, 17]
 figure=[7]
 finally=[12]
 find=[12]
 first=[12]
 follow=[15]
 for=[8, 9]
 found=[15]
 from=[19]
 front=[6]
 gold=[9]
 google=[16]
 gradually=[11]
 had=[3]
 have=[9]
 hobby=[11]
 how=[7]
 however=[3]
 html=[12]
 hunger=[9]
 i=[3, 4, 5, 8, 10, 11, 12, 13, 15, 19]
 if=[8]
 imagination=[10]
 in=[1, 4, 6, 9, 11, 16, 17]
 innate=[3]
 innovation=[2]
 interest=[5]
 into=[17]
 introductory=[4, 16]
 is=[1]
 it=[1, 5, 9, 13]
 junior=[4]
 just=[9]
 keep=[2]
 knowledge=[18]
 lap=[9]
 later=[12]
 learned=[12]
 learning=[2]
 like=[8]
 limit=[2]
 line=[6]
 lives=[19]
 make=[13, 16]
 many=[6, 11]
 may=[10]
 me=[3, 8, 17, 18]
 meanwhile=[15]
 medal=[9]
 meter=[9]
```





Q6： Design and create a hospital information system with the following scenarios.

● Register a new patient.

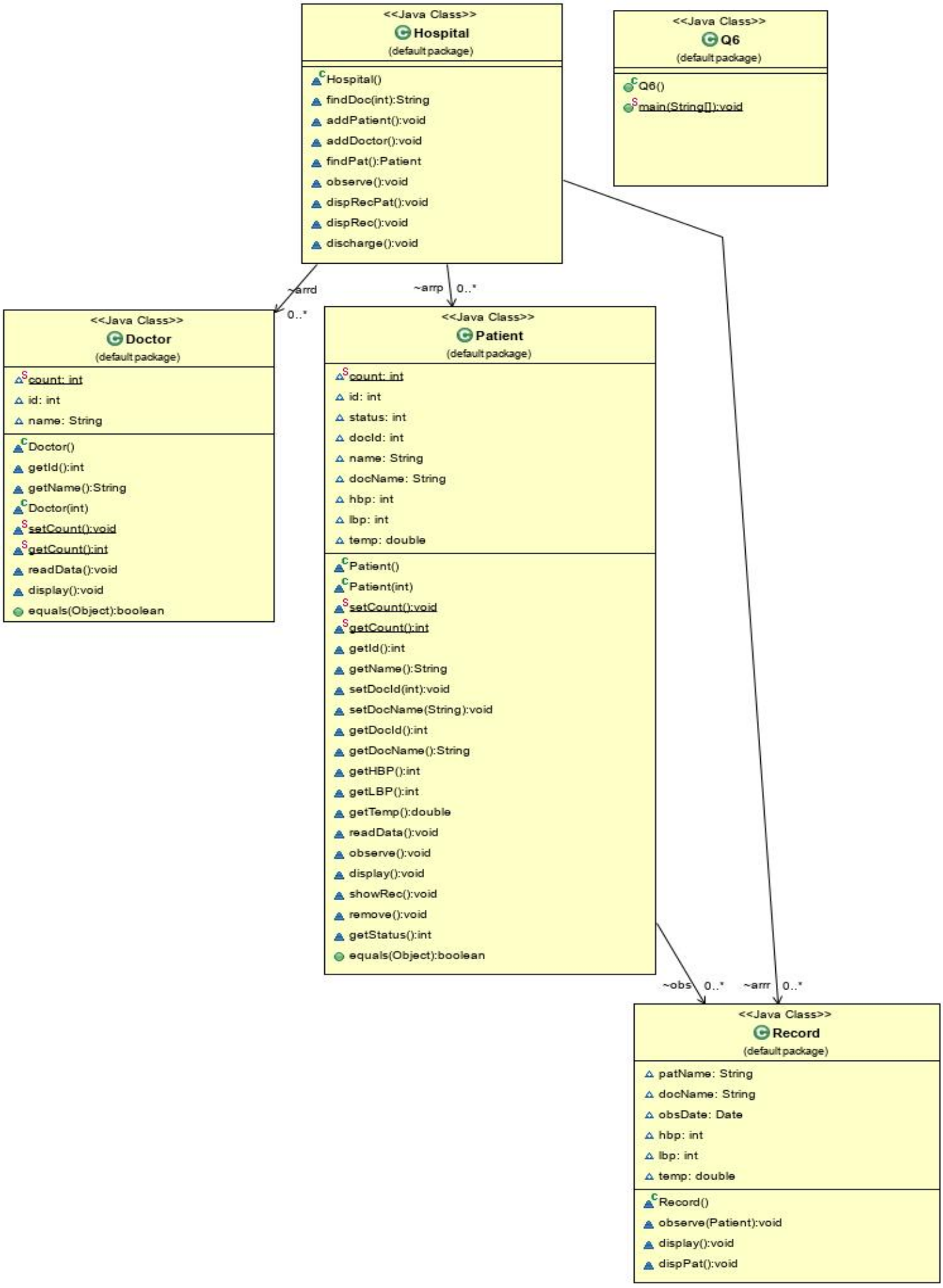
● Each patient is assigned to one doctor, but a doctor can have any number of patients. Patients check in to the hospital and assigned a doctor if they don't already have one.

● While in the hospital, doctors record various observations about each patient at various times. Examples of observations are blood pressure and temperature. Record test results for a patient.

● The hospital keeps track of all the observations for a given patient until they check out of the hospital. Obtain all of a patient's information given the social security number.

NOTE: Patients id will be auto generated.

UML Diagram:


<<Java Class>>
Hospital
(default package)
Hospital()
findDoc(int):String
addPatient():void
addDoctor():void
findPat():Patient
observe():void
dispRecPat():void
dispRec():void
discharge():void
<<Java Class>>
Q6
(default package)
Q6()
main(String[]):void
~arrd
0..*
~arrp
0..*
<<Java Class>>
Doctor
(default package)
count: int
id: int
name: String
Doctor()
getId():int
getName():String
Doctor(int)
setCount():void
getCount():int
readData():void
display():void
equals(Object):boolean
<<Java Class>>
Patient
(default package)
count: int
id: int
status: int
docId: int
name: String
docName: String
hbp: int
lbp: int
temp: double
Patient()
Patient(int)
setCount():void
getCount():int
getId():int
getName():String
setDocId(int):void
setDocName(String):void
getDocId():int
getDocName():String
getHBP():int
getLBP():int
getTemp():double
readData():void
observe():void
display():void
showRec():void
remove():void
getStatus():int
equals(Object):boolean
~obs
0..*
~arrr
0..*
<<Java Class>>
Record
(default package)
patName: String
docName: String
obsDate: Date
hbp: int
lbp: int
temp: double
Record()
observe(Patient):void
display():void
dispPat():void

Output :

```
atanu@dell:~/Documents/OOP Adv Lab/java_assignments1/q6$ java Q6
Enter choice
1. Admit patient
2. Enter doctor
3. Observe a patient
4. Discharge a patient
5. Display observation records of a patient
6. Display all observation records
7. Exit
1
No Doctors in hospital...Enter doctors first.
Enter choice
1. Admit patient
2. Enter doctor
3. Observe a patient
4. Discharge a patient
5. Display observation records of a patient
6. Display all observation records
7. Exit
2
Enter name
Souren Dey
Doctor id : 20001
Dr. Souren Dey
Enter choice
1. Admit patient
2. Enter doctor
3. Observe a patient
4. Discharge a patient
5. Display observation records of a patient
6. Display all observation records
7. Exit
2
Enter name
Koustav Roy
Doctor id : 20002
Dr. Koustav Roy
Enter choice
1. Admit patient
2. Enter doctor
3. Observe a patient
4. Discharge a patient
5. Display observation records of a patient
6. Display all observation records
7. Exit
1
Enter name
Atanu Ghosh
Enter doctor id you want to be assigned
20001
Patient id : 10001
```

```
Enter name
Atanu Ghosh
Enter doctor id you want to be assigned
20001
Patient id : 10001
Name : Atanu Ghosh
Allocated doctor : Dr. Souren Dey
Enter choice
1. Admit patient
2. Enter doctor
3. Observe a patient
4. Discharge a patient
5. Display observation records of a patient
6. Display all observation records
7. Exit
1
Enter name
Bablu Ghosh
Enter doctor id you want to be assigned
20001
Patient id : 10002
Name : Bablu Ghosh
Allocated doctor : Dr. Souren Dey
Enter choice
1. Admit patient
2. Enter doctor
3. Observe a patient
4. Discharge a patient
5. Display observation records of a patient
6. Display all observation records
7. Exit
1
Enter name
Rita Ghosh
Enter doctor id you want to be assigned
20002
Patient id : 10003
Name : Rita Ghosh
Allocated doctor : Dr. Koustav Roy
Enter choice
1. Admit patient
2. Enter doctor
3. Observe a patient
4. Discharge a patient
5. Display observation records of a patient
6. Display all observation records
7. Exit
3
Enter patient code
10003
Enter systolic blood pressure
```

```
Enter patient code
10003
Enter systolic blood pressure

120
Enter diastolic blood pressure

80
Enter temperature

99
Enter choice
1. Admit patient
2. Enter doctor
3. Observe a patient
4. Discharge a patient
5. Display observation records of a patient
6. Display all observation records
7. Exit
5
Enter patient code
10003
Patient id : 10003
Patient name : Rita Ghosh
Observing Doctor : Dr.Koustav Roy
Sl.No.  Date    Time    Systolic pressure       Diastolic pressure      Temperature
1       07/05/2021      09:03:08        120     80      99.0
Enter choice
1. Admit patient
2. Enter doctor
3. Observe a patient
4. Discharge a patient
5. Display observation records of a patient
6. Display all observation records
7. Exit
6
Sl.No.  Patient Doctor  Date    Time    Systolic pressure       Diastolic pressure      Temperature
1       Rita Ghosh      Koustav Roy     07/05/2021      09:03:08        120     80      99.0
Enter choice
1. Admit patient
2. Enter doctor
3. Observe a patient
4. Discharge a patient
5. Display observation records of a patient
6. Display all observation records
7. Exit
4
Enter patient code
10001
Enter choice
1. Admit patient
```

```
Enter patient code
10003
Patient id : 10003
Patient name : Rita Ghosh
Observing Doctor : Dr.Koustav Roy
Sl.No.  Date    Time    Systolic pressure       Diastolic pressure      Temperature
1       07/05/2021      09:03:08        120     80      99.0
Enter choice
1. Admit patient
2. Enter doctor
3. Observe a patient
4. Discharge a patient
5. Display observation records of a patient
6. Display all observation records
7. Exit
6
Sl.No.  Patient Doctor  Date    Time    Systolic pressure       Diastolic pressure      Temperature
1       Rita Ghosh      Koustav Roy     07/05/2021      09:03:08        120     80      99.0
Enter choice
1. Admit patient
2. Enter doctor
3. Observe a patient
4. Discharge a patient
5. Display observation records of a patient
6. Display all observation records
7. Exit
4
Enter patient code
10001
Enter choice
1. Admit patient
2. Enter doctor
3. Observe a patient
4. Discharge a patient
5. Display observation records of a patient
6. Display all observation records
7. Exit
4
Enter patient code
10001
Enter choice
1. Admit patient
2. Enter doctor
3. Observe a patient
4. Discharge a patient
5. Display observation records of a patient
6. Display all observation records
7. Exit
7
Exitting
atanu@dell:~/Documents/OOP Adv Lab/java_assignments1/q6$
```