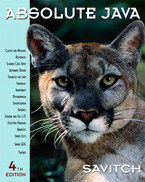
** COSC 1320**

**Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Seat # \_\_\_**

**Estimated Hours 1**

**Actual Hours**

**TA (Textual Analysis) for UML USE CASE Diagram MODEL**

**Programming Assignment 1:**

**(10 points)**

**(Due date found in the COSC 1320 BB Calendar!)**

**You must use Microsoft WORD program. Insert the TEMPLATEs for Actors and Use Cases in this Requirements Document.**

*Please have Step 1, 2, 3, and 4 in this order.*

PLEASE use the “TA for UML USE CASE Diagram MODEL Movie Company System.doc” AS TEMPLATE.

(Do not Show STEP 1,…; do not change Line Numbers; do not change Number of Pages)

Any DIAGRAM that is NOT the result of CUT and PASTE

WILL BE IGNORED. (YOU WILL GET ZERO POINTS)

**Requirements Analysis**: Construct the **UML Use Case Diagram** MODEL

**Requirements Analysis**: Construct the **UML USE CASE** Diagram MODEL

**ProgrammingAssignment1** **APPLICATION** to automate **adding**, **deleting**, and **displaying** the list of employees in a hospital.

HospitalEmployee

Employee

The set of **Class**es define these employees of a hospital: hospital employee, doctor, nurse, administrator, surgeon, receptionist, and janitor. **MUST use inheritance** in creating these **Class**es. You can use **arrays** to store objects of the same **Class**, you can assume MAX\_SIZE = 5 or **one array** to store all objects of all the **Class**es, you can assume MAX\_SIZE = 20.

Employee1:MAX\_SIZE: int

Employee1: empArr: Array

Janitor

Receptionist

Surgeon

Administrator

Nurse

Doctor

(you are **not** to import java.util.\*; - failure to conform will result in losing 20 points; you are **not** to use **JCL** **ArrayList** or **LinkedList** or **Vector** Classes - failure to conform will result in losing 40 points; you are **to** use an C array - failure to conform will result in losing 30 points)

These employees are initially read and added from the **“Programming Assignment 1 Data.txt”** and are also **saved in same file** when your **ProgrammingAssignment1** **APPLICATION** terminates.

Employee: readFile(String file):void

Your **ProgrammingAssignment1** **APPLICATION** must **read** in the data file (FILE INPUT) into your **ProgrammingAssignment1** **APPLICATION** memory (**please DO NOT read from File and write to File skipping reading into the program memory**).

Your **ProgrammingAssignment1** **APPLICATION** must allow the user to **delete** a hospital employee, doctor, nurse, administrator, surgeon, receptionist, and janitor given the role and the name.

Employee: deleteEmployee(char r , String n):void

Your **ProgrammingAssignment1** **APPLICATION** must allow the user to **add** a hospital employee, doctor, nurse, administrator, surgeon, receptionist, and janitor.

Employee: addEmployee(Employee obj): void

Your **ProgrammingAssignment3** **APPLICATION** must allow the user to **display** **the hospital employees** in the format given below.

Employee: addEmployee(Employee obj): void

Your **ProgrammingAssignment3** **APPLICATION** must allow the user to **save** the hospital employees before it exits from the **ProgrammingAssignment3** **APPLICATION** from the memory to the **SAME** data file (FILE OUTPUT) **“Programming Assignment 1 Data.txt”**.

Employee: writeFile(String file): void

Use **Constructors** to automatically initialize the **instance variables** that MUST be declared **private** - failure to make ALL **member variables** **private** will result in losing 20 points. Appropriate accessor and mutator **methods** for each **private** **instance variable** must be created.

Add (**overwrite**) the **methods** to display (must use **toString** method and super - failure to use **toString** will result in losing 10 points; - failure to use super will result in losing 10 points).

**MUST use inheritance** in creating these **Class**es - failure to use **inheritance** will result in losing 20 points).

A sample input file **“Programming Assignment 1 Data.txt”** follows:

h Vito 123

d Michael 234 Heart

n Sonny 456 6

a Luca 567 Business

r Tom 678 Talking Y

j Anthony 789 Maintenance N

d Nicos 891 Bone

Employee: name: String

Employee: role: char

The h stands for hospital employee role, Vito for **name** (only one string), and 123 is its hospital employee **number**.

HospitalEmployee1: Surgeon()

HospitalEmployee2: Surgeon(char , String , int , String , char)

HospitalEmployee3: deleteEmployee():void

HospitalEmployee4: toString():String

HospitalEmployee5: equals(Employee obj):boolean

Employee: idNumber: int

The d stands for doctor, Michael for **name**, 234 for **number**, and Heart for **specialty**.

Doctor2: Doctor()

Doctor3: Doctor(char , String , int , String)

Doctor4: deleteEmployee(): void

Doctor5: setSpec(String):void

Doctor6: getSpec():String

Doctor7: toString():String

Doctor8: equals(Employee obj):boolean

Doctor1: specialty: String

The s stands for surgeon, Vincent for **name**, 345 for **number**, Brain for **specialty**, and Y for **operating**.

Surgeon3: Surgeon()

Surgeon4: Surgeon(char , String , int , String , char)

Surgeon5: deleteEmployee():void

Surgeon6: setSpec(String):void

Surgeon7: getSpec():String

Surgeon8: setOpStat(String):void

Surgeon9: getOpStat():String

Surgeon10: toString():String

Surgeon11: equals(Employee obj):boolean

Surgeon1: specialty: String

Surgeon2: operating: char

The n stands for nurse, Sonny for **name**, 456 for **number**, and 6 for **numpatients**.

Nurse2: Nurse ()

Nurse3: Nurse (char , String , int , String)

Nurse4: deleteEmployee():void

Nurse5: setPtCnt(int):void

Nurse6: getPtCnt ():int

Nurse7: toString():String

Nurse8: equals(Employee obj):boolean

Nurse1: numPatients: int

The a stands for administrator, Luca for **name**, 567 for **number**, and Business for **department**.

Administrator2: Administrator ()

Administrator3: Administrator (char , String , int , String)

Administrator4: deleteEmployee():void

Administrator5: setPtCnt(int):void

Administrator6: getPtCnt ():int

Administrator7: toString():String

Administrator8: equals(Employee obj):boolean

Administrator1: department: String

The r stands for receptionist, Tom for **name**, 678 for **number**, Talking for **department**, and Y for **answering**.

Receptionist1: department: String

Receptionist2: answering: char

Receptionist3: Receptionist ()

Receptionist4: Receptionist (char , String , int , String , char)

Receptionist5: deleteEmployee():void

Receptionist6: setDept (String):void

Receptionist7: getDept ():String

Receptionist8: setOpStat(String):void

Receptionist9: getOpStat():String

Receptionist10 toString():String

Receptionist11: equals(Employee obj):boolean

The j stands for janitor, Anthony for **name**, 789 for **number**, Maintenance for **department**, and Y for **sweeping**.

Janitor1: department: String

Janitor3: Surgeon()

Janitor4: Surgeon(char , String , int , String , char)

Janitor5: deleteEmployee():void

Janitor6: setSpec(String):void

Janitor7: getSpec():String

Janitor8: setOpStat(String):void

Janitor9: getOpStat():String

Janitor10: toString():String

Janitor11: equals(Employee obj):boolean

Janitor2: sweeping: char

Format to **display** the Hospital Employees use the following format (**must use** the **toString**() **method**):

**The Hospital has the following employees:**

**Hospital Employees: 1**

**Name: Vito Employee Number: 123**

**Doctors: 2**

**Name: Michael Employee Number: 234 Specialty: Heart**

**Name: Nicos Employee Number: 891 Specialty: Bone**

**Surgeons: 0**

**Nurses: 1**

**Name: Sonny Employee Number: 456 Number of Patients: 6**

**Administrators: 1**

**Name: Luca Employee Number: 567 Department: Business**

**Receptionists: 1**

**Name: Tom Employee Number: 678 Department: Talking Answering: Y**

**Janitors: 1**

**Name: Anthony Employee Number: 789 Department: Maintenance Sweeping: N**

**Total number of Employees: 7**

Create a **ProgrammingAssignment1.java** that contains the **main method**.

|  |
| --- |
| Main |
| - View view |
| + main() |

|  |
| --- |
| View |
| Scanner keyboard |
| + showMenu():void  + enterEmployeeData():void  + deleteEmployeeData():void  + displayEmployeeData():void  + readDataFromFile():void  + writeDataToFile():void |

|  |
| --- |
| HospitalEmployee |
|  |
| + HospitalEmployee():void  + HospitalEmployee(char, String, int):void  + deleteEmployee(String , char): void  + toString():String  + equals(Employee):boolean |

|  |
| --- |
| Doctor |
| - specialty: String |
| + Doctor():void  + Doctor(char, String, int, String):void  + deleteEmployee(String , char): void  + getSpec():char  + setSpec(char):void  + toString():String  + equals(Employee):boolean |

|  |
| --- |
| Employee |
| - MAX\_SIZE: int  - empArr: Array  - employeeCtr: int  - name: String  - idNumber: int  - role: char |
| + Employee():void  + Employee(char, String, int):void  + addEmployee(Employee):void  + deleteEmployee(String , char): void  + displayEmployees():void  + readFile(String): void  + writeFile(String): void  + getRole():char  + getName(): String  + getID():int  + setRole(char):void  + setName(String):void  + setID(int):void  + toString():String  + equals(Employee):boolean |

|  |
| --- |
| Surgeon |
| - specialty: String |
| + Surgeon ():void  + Surgeon (char, String, int, String, char):void  + deleteEmployee(String , char): void  + setOpStat ():char  + setSpec (): String  + getSpec (char):void  + getOpStat (String):void  + toString():String  + equals(Employee):boolean |

|  |
| --- |
| Administrator |
| - department: String |
| + Administrator():void  + Administrator(char, String, int, String):void  + deleteEmployee(String , char): void  + setDept():String  + getDept(String):void  + toString():String  + equals(Employee):boolean |

|  |
| --- |
| Receptionist |
| - department: String  - answering: char |
| + Receptionist():void  + Receptionist(char, String, int, String, char):void  + deleteEmployee(String , char): void  + setDept():String  + getDept(String):void  + setWorkStat():char  + getWorkStat(char):void  + toString():String  + equals(Employee):boolean |

|  |
| --- |
| Janitor |
| - department: String  - sweeping: char |
| + Janitor():void  + Janitor(char, String, int, String, char):void  + deleteEmployee(String , char): void  + setDept():String  + getDept(String):void  + setWorkStat():char  + getWorkStat(char):void  + toString():String  + equals(Employee):boolean |

|  |
| --- |
| Nurse |
| - numOfPatients: int |
| + Nurse ():void  + Nurse (char, String, int, int):void  + deleteEmployee(String , char): void  + setPtCnt ():int  + getPtCnt (int):void  + toString():String  + equals(Employee):boolean |