BINUS University

Academic Career: ***Undergraduate / ~~Master~~ / ~~Doctoral~~ *)***			**Class Program:** ***~~International~~/Regular/~~Smart Program~~/Global Class*)***
☐ **Mid Exam** ☑ **Final Exam** ☐ **Short Term Exam** ☐ **Others Exam : _____**			**Term : ~~Odd~~/Even/~~Short~~ *)**
☑ **Kemanggisan** ☑ **Alam Sutera** ☐ **Bekasi** ☐ **Senayan** ☐ **Bandung** ☐ **Malang**			**Academic Year :** **2019 / 2020**
Faculty / Dept.	: School of Computer Science	Deadline	Day / Date : Tuesday / Jul 07th, 2020 Time : 13:00 - 16:20
Code - Course	: COMP6175 – Object Oriented Programming	Class	: All Classes
Lecturer	: Team	Exam Type	: Online
*) *Strikethrough the unnecessary items*			
The penalty for CHEATING is DROP OUT!!!			

The total duration of this exam is 200 minutes, including downloading the questions and uploading the answers. Please use the time provided wisely.

I. Case (100%)

XYZ University is designing their Information System. They are classifying all people in their university as a person. So a person can be a student or an employee. And an employee can be a faculty member or an administrator. A student is a person that is studying in the university. A faculty is the one that is teaching in the university. And the one that is administering daily activities in the university is an administrator.

This is the design for XYZ University Information System:

- Class **Orang** that has attributes such as name, address, phone number, and email.
- Class **Orang** is derived into class **Mahasiswa** and class **Karyawan**.
- Class **Mahasiswa** has an additional attribute to represent the year of the student.
- Class **Karyawan** has additional attributes to represent the office location, the salary, and the date when they started working.
- Class **Dosen** and class **Administrator** are derived from class **Karyawan**.
- Class **Dosen** has more additional attributes to represent the teaching hours and the academic level.
- Class **Administrator** has one additional attribute to represent the title.
- Override the method **toString** in each class to return the class name and the name attribute of that class.
- Class **Orang** and **Karyawan** can't be instantiated.

Each user in the system is considered logged in into the system and they each can do a few things when they are logged in. So you need to create a list of menu for each user.

Verified by,

Fidelson Tanzil (D5542) and sent to Department on May 29, 2020

- Compulsory menu for every person in class **Mahasiswa** is:
  1. Profile
  2. Change address
  3. Change phone number
  4. Change email
  5. Change user

  With descriptions as follow:
  - Profile is to show all the attributes of the user.
  - Change address is to change the user's address.
  - Change phone number is to change the user's phone number.
  - Change email is to change the user's email address.
  - Change user is to change to another user.

- Compulsory menu for every person in class **Dosen** is:
  1. Profile
  2. Change address
  3. Change phone number
  4. Change email
  5. Change user

  With descriptions as follow:
  - Profile is to show all the attributes of the user.
  - Change address is to change the user's address.
  - Change phone number is to change the user's phone number.
  - Change email is to change the user's email address.
  - Change user is to change to another user.

- Compulsory menu for every person in class **Administrator** is:
  1. Profile
  2. Add Mahasiswa
  3. Add Dosen
  4. Add Administrator
  5. List all Mahasiswa
  6. List all Dosen
  7. List all Administrator
  8. List all user
  9. Change user

  With descriptions as follow:
  - Profile is to show all the attributes of the user.
  - Add Mahasiswa is to add another user as a student
  - Add Dosen is to add another user as a faculty member.
  - Add Administrator is to add another user as an administrator.
  - List all Mahasiswa is to print down the name and the year of all students.
  - List all Dosen is to print down the name, office location, and the academic level of all faculty members.
  - List all Administrator is to print down the name, office location, and the title of all the administrators.
  - List all user is to print down the class name and the name of the user. (Hint: Use the method **toString**)
  - Change user is to change to another user.

Verified by,

Fidelson Tanzil (D5542) and sent to Department on May 29, 2020

Additional information for XYZ University Information System:

- Year for students are 2015, 2016, 2017, 2018, 2019, and 2020.
- Office locations for employees are Kemanggisan, Alam Sutra, Bekasi, Bandung, and Malang.
- Academic levels for faculty members are Asisten Ahli, Lektor, Lektor Kepala, and Professor. (https://id.wikipedia.org/wiki/Dosen#Jabatan/Pangkat_Dosen)
- Titles for administrators are Staff, Supervisor, Manager, Kepala Divisi, and Direksi.
- Phone number must be validated to contain only numbers. (No characters or special symbols)
- Email doesn't have to be validated to a standard email form.
- Date when an employee starts working can user the class **Calendar** provided by Java.
- Don't forget to implement the use of **inheritance**, **polymorphism**, **class Vector** or **class ArrayList** and **multi-thread concept**.
- **An example to multi-thread programming and how to change user has been provided below**.

MainThread.java

```
import java.util.*;
import java.util.concurrent.locks.*;

public class MainThread {
    private Lock objLock = new ReentrantLock();
    private Condition changeUser = objLock.newCondition();
    private Scanner sc;

    public static void main(String[] args) {
        new MainThread();
    }

    public MainThread() {
        sc = new Scanner(System.in);

        UserTask ut1 = new UserTask("Adi");
        UserTask ut2 = new UserTask("Budi");

        Thread t1 = new Thread(ut1);
        Thread t2 = new Thread(ut2);

        t1.start();
        t2.start();

        try {
            t1.join();
            t2.join();
        } catch(InterruptedException e) {
            System.out.println(e);
        }

        System.out.println("Main Thread Finished\n\n");
        sc.close();
    }

    public class UserTask extends User implements Runnable{
        public UserTask(String nama) {
            super(nama);
        }

        public void run() {
            int choice;
            try {
                objLock.lock();
                do {
                    System.out.println("1. Current User: " + getName() );
                    System.out.println("2. Change User");
```

Verified by,

Fidelson Tanzil (D5542) and sent to Department on May 29, 2020

```
                System.out.println("3. Exit");
                System.out.print("Enter Choice >> ");

                choice = sc.nextInt();
                switch(choice) {
                case 1:
                    System.out.println();
                    break;

                case 2:
                    if(Thread.activeCount() > 2) {
                        System.out.println();
                        changeUser.signal();
                        changeUser.await();
                    } else {
                        System.out.println("Can't change user.\n");
                    }
                    break;
                case 3:
                    System.out.println( getName() + " logging out\n");
                    changeUser.signal();
                    break;
                default:
                    System.out.println("Wrong choice!!\n\n");
                }
            } while( choice != 3 );
        } catch(InterruptedException e) {
            System.out.println(e);
        } finally {
            objLock.unlock();
        }
    }
}
```

MainThreadUsingExecutor.java
//This is an implementation of thread execution using thread pool

```
import java.util.Scanner;
import java.util.concurrent.*;
import java.util.concurrent.locks.*;

public class MainThreadUsingExecutor {
    private Lock objLock = new ReentrantLock();
    private Condition changeUser = objLock.newCondition();
    private Scanner sc;

    public static void main(String[] args) {
        new MainThreadUsingExecutor();
    }

    public MainThreadUsingExecutor() {
        sc = new Scanner(System.in);
        ExecutorService es = Executors.newFixedThreadPool(2);

        es.execute(new UserTask("Aldo"));
        es.execute(new UserTask("Bryan"));
        es.shutdown();

        try {
            es.awaitTermination(24L, TimeUnit.HOURS);
        } catch(InterruptedException e) {
            System.out.println(e);
        }
```

Verified by,

Fidelson Tanzil (D5542) and sent to Department on May 29, 2020

```
        System.out.println("Main Thread Finished\n\n");
        sc.close();
    }

    public class UserTask extends User implements Runnable{
        public UserTask(String nama) {
            super(nama);
        }

        public void run() {
            int choice;
            try {
                objLock.lock();
                do {
                    System.out.println("1. Current User: " + getName() );
                    System.out.println("2. Change User");
                    System.out.println("3. Exit");
                    System.out.print("Enter Choice >> ");

                    choice = sc.nextInt();
                    switch(choice) {
                    case 1:
                        System.out.println();
                        break;
                    case 2:
                        if(Thread.activeCount() > 2) {
                            System.out.println();
                            changeUser.signal();
                            changeUser.await();
                        } else {
                            System.out.println("Can't change user.\n");
                        }
                        break;
                    case 3:
                        System.out.println( getName() + " logging out\n");
                        changeUser.signal();
                        break;
                    default:
                        System.out.println("Wrong choice!!\n\n");
                    }
                } while( choice != 3 );
            } catch(InterruptedException e) {
                System.out.println(e);
            } finally {
                objLock.unlock();
            }
        }
    }
}
```

User.java

```
public class User {
    private String name;

    public User(String name) {
        this.name = name;
    }

    public String getName() {
        return name;
    }
}
```

-- Good Luck --

Verified by,

Fidelson Tanzil (D5542) and sent to Department on May 29, 2020