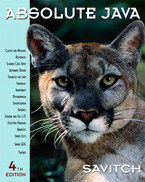
** COSC 1320**

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

**Estimated Hours 2**

**Actual Hours**

**TA (Textual Analysis) for UML MVC CLASS Diagram MODEL**

**Programming Assignment 2:**

**(10 points)**

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

**You must use Microsoft WORD program. Insert the TEMPLATEs for Class, Attribute, etc. in this Requirements Document.**

*Please have Step 1, 2, 3, 4, and 5 in this order.*

PLEASE use the “TA for UML MVC Class 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

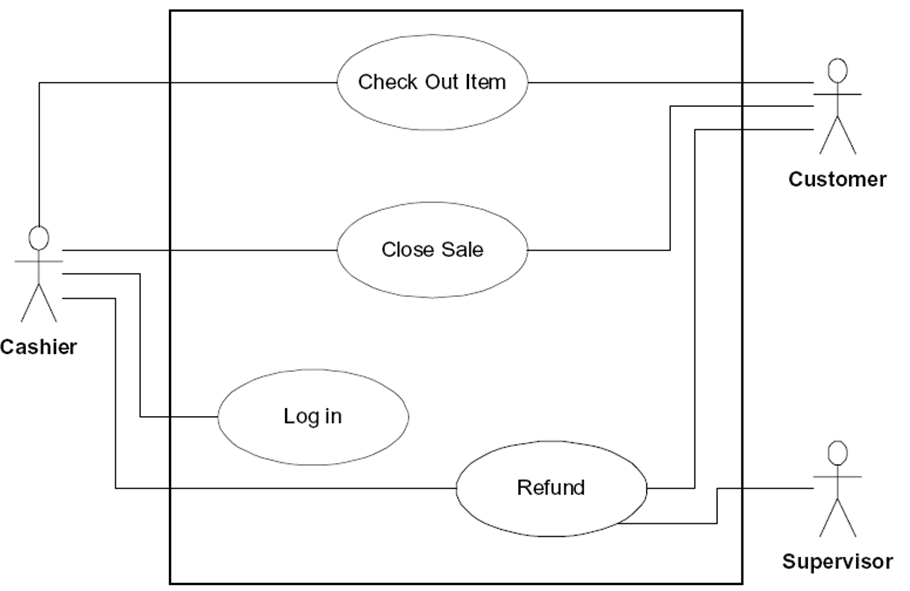
WILL BE IGNORED. (YOU WILL GET ZERO POINTS)

**Requirements Analysis**: Construct the **UML MVC CLASS** Diagram MODEL

**Analysis and Design**: Construct the **UML MVC CLASS** Diagram MODEL

ProgrammingAssignment2 APPLICATION for ANIMATING an electrical CIRCUIT using SWING.

ElectricalCircuit

Write an **application** that controls an electrical circuit that has a power source, a switch, and a light bulb. The user wants to control the names of the Labels, thus they must be read from a data file named “**Programming Assignment 2 Data.txt**”.

LightBulb

SwitchBoard

**A1: User**

A sample input file **“Programming Assignment 2 Data.txt”** follows:

Animation of an Electrical Circuit

Power Supply

Switch

Light bulb

SwitchBoard: 1 – openButton: JButton

SwitchBoard: 2 – closeButton: JButton

You will have two buttons, labeled Open and Close. When you click on Close (see that the switch shows that it is closed - redraw) the light bulb should turn on. When you click on Open (see that the switch shows that it is open - redraw) the light bulb should turn off.

SwitchBoard: 2 – closeButton: JButton

The title of your window must be “Programming Assignment 2”.

Create a **ProgrammingAssignment2.java** that contains the **main method**.

|  |
| --- |
| Main |
|  |
| + main(String): void //1 |

|  |
| --- |
| Controller |
| - view: View  - circuit: ElectricalCircuit |
| + Controller(): void //1 |

|  |
| --- |
| View |
| - WINDOW\_WIDTH:int //1  - WINDOW\_HEIGHT: int //2 |
| + View():void //1 |

|  |
| --- |
| ElectricalCircuit |
| -JTextField textField:JTextField //1  - state: boolean //2  - top: int //3  - bottom: int //4  - left: int //5  - right: int //6  - midX: int //7  - midY: int //8  - wires: Graphics //9 |
| + ElectricalSocket(): void //1  + paintComponent(Graphics g):void //2  + drawSocket():void //3  + openGate():void //4  + closeGate():void //5 |

|  |
| --- |
| SwitchBoard |
| - textField: JTextField //1  - state: String //2  - bulb: LightBulb //3  - circuit: ElectricalCircuit //4 |
| + SwitchBoard (): void //1  + actionPerformed(ActionEvent):void //2  + printStatus(ActionEvent):void //3 |

|  |
| --- |
| LightBulb |
| - centerX: int //1  - centerY: int //2  - width: int //3  - height: int //4  - flag: String //5 |
| + LightBulb(): void //1  + paint(Graphics g):void //2  + flipState(ActionEvent):void //3  + getWidth ():int //4  + getHeight ():int //5 |

|  |
| --- |
| Main |
|  |
| + main(String): void //1 |

public static void main(String[] args){

This method kicks off the Controller to start the program.

|  |
| --- |
| Controller |
| - view: View  - circuit: ElectricalCircuit |
| + Controller(): void //1 |

}

Controller(){

Creates the View, SwitchBoard and ElectricalCircuit.

Once they’re configured, the switches and circuits are added to the View.

|  |
| --- |
| View |
| - WINDOW\_WIDTH:int //1  - WINDOW\_HEIGHT: int //2 |
| + View():void //1 |

}

View(){

Initializes the window settings (title, width, height, layout, and exit functionality).

}

|  |
| --- |
| SwitchBoard |
| - textField: JTextField //1  - state: String //2  - bulb: LightBulb //3  - circuit: ElectricalCircuit //4 |
| + SwitchBoard (): void //1  + actionPerformed(ActionEvent):void //2  + printStatus(ActionEvent):void //3 |

SwitchBoard(){

Creates a bulb (LightBulb), circuit (ElectricalCircuit), and buttonPanel (JPanel) objects.

Creates a textField (JTextField) to write the current state of the circuit.

}

public void actionPerformed(ActionEvent e) {

Tries to call printStatus(ActionEvent e) and passes the event, otherwise, the textField displays an error message.

}

public void printStatus(ActionEvent e){

Reads the action command that took place and lights the bulb or turns it off.

Also, prints the state of the bulb to textField.

}

|  |
| --- |
| LightBulb |
| - centerX: int //1  - centerY: int //2  - width: int //3  - height: int //4  - flag: String //5 |
| + LightBulb(): void //1  + paint(Graphics g):void //2  + flipState(ActionEvent):void //3  + getWidth ():int //4  + getHeight ():int //5 |

LightBulb(){

Calls the parent JPanel.

}

public void paint(Graphics g){

Depending on the state of the flag, the bulb is painted LIGHT\_GRAY or YELLOW.

}

public void flipState(ActionEvent e){

Changes the flag to the actionCommand that was passed

Repaints the bulb.

}

public int getWidth(){

return width;

}

public int getHeight(){

return height;

}

|  |
| --- |
| ElectricalCircuit |
| -JTextField textField:JTextField //1  - state: boolean //2  - top: int //3  - bottom: int //4  - left: int //5  - right: int //6  - midX: int //7  - midY: int //8  - wires: Graphics //9 |
| + ElectricalSocket(): void //1  + paintComponent(Graphics g):void //2  + drawSocket():void //3  + openGate():void //4  + closeGate():void //5 |

ElectricalCircuit(){

Sets the layout and renders the visibility to true.

}

public void paintComponent(Graphics g) {

Casts g as a Graphics2D object and uses drawLine(x , y , x2 , y2) to draw the circuit.

If state is set to true, then closeGate() is called. Otherwise, openGate() is called.

}

public void drawSocket(Graphics2D g){

Draws the bulb socket.

Poor implementation. Didn’t know how to lay it all out properly.

}

public void openGate(){

Tries to redraw the gate line to complete the circuit.

}

public void closeGate(){

Tries to redraw the gate line to break the circuit.

}