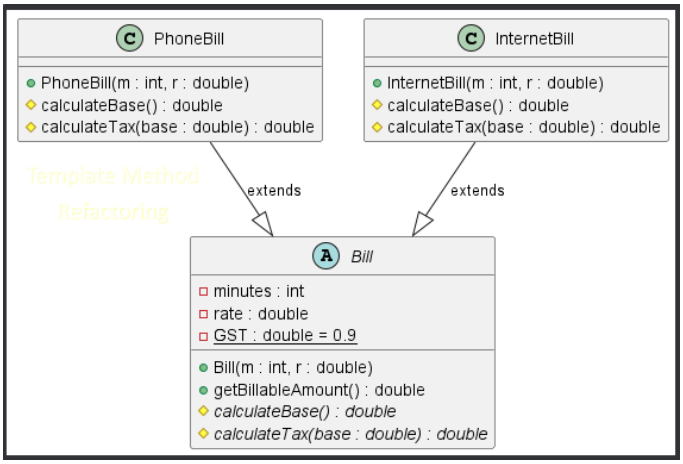
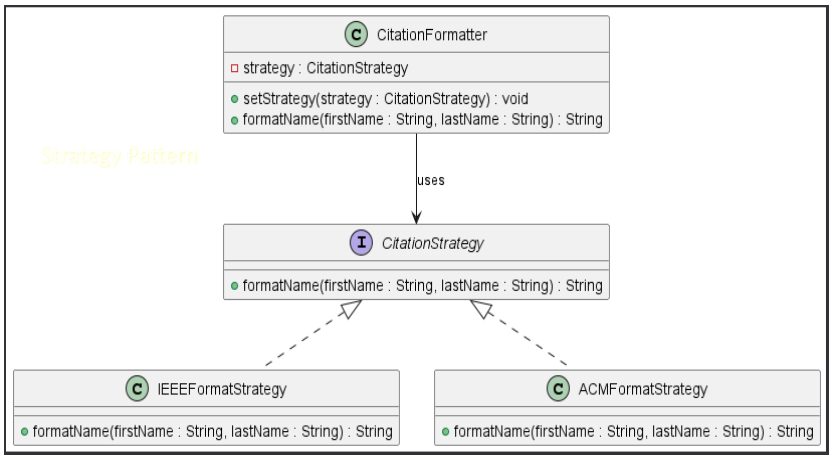
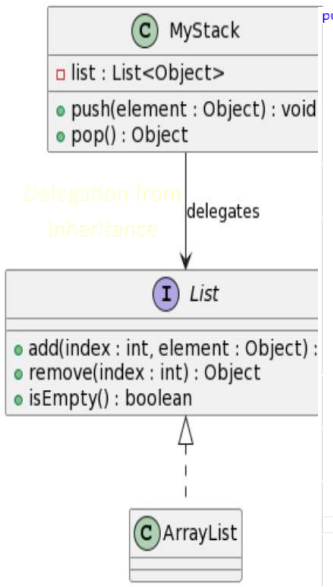


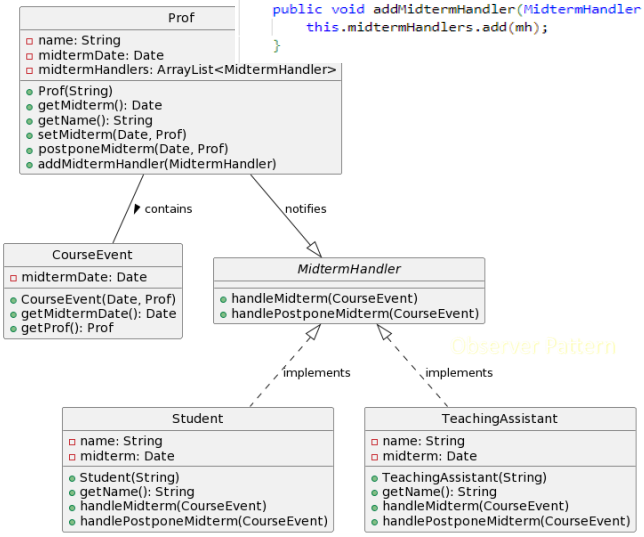



```java
// Bill.java
public abstract class Bill {
    protected int minutes;
    protected double rate;
    protected static final double GST = 0.9;

    public Bill(int m, double r) {
        minutes = m;
        rate = r;
    }

    // Template method
    public final double getBillableAmount() {
        double base = calculateBase();
        double tax = calculateTax(base);
        return base + tax;
    }

    // Abstract methods for the steps that vary
    protected abstract double calculateBase();

    protected abstract double calculateTax(double base);
}
```




```java
public class Prof {
    private String name;
    private Date midtermDate;
    private ArrayList<MidtermHandler> midtermHandlers;

    public Prof(String aName) {
        this.name = aName;
        this.midtermHandlers = new ArrayList<MidtermHandler>();
    }

    public Date getMidterm() {
        return this.midtermDate;
    }

    public String getName() {
        return this.name;
    }

    public void setMidterm(Date date, Prof prof) {
        this.midtermDate = date;
        CourseEvent event = new CourseEvent(this.midtermDate, prof);
        for(MidtermHandler s: this.midtermHandlers){
            s.handleMidterm(event);
        }
    }

    public void postponeMidterm(Date date, Prof prof){
        this.midtermDate = date;
        CourseEvent event = new CourseEvent(date, prof);
        for(MidtermHandler s: this.midtermHandlers){
            s.handlePostponeMidterm(event);
        }
    }

    public void addMidtermHandler(MidtermHandler mh){
        this.midtermHandlers.add(mh);
    }
```

```java
public class PasswordValidator {

    Run | Debug
    public static void main(String[] args) {
        try {
            Scanner scanner = new Scanner(System.in);
            System.out.print(s:"Enter your password: ");
            String password = scanner.nextLine();

            if (isValidPassword(password)) {
                System.out.println(x:"Password is valid.");
            } else {
                System.out.println(x:"Password is invalid.");
            }
        } catch (Exception e) {
            System.out.println("An error occurred: " + e.getMessage());
        }
    }

    private static boolean isValidPassword(String password) {
        String regex = "^(?=.*[a-z])(?=.*[A-Z])(?=.*\\d)(?=.*[!@#\\$%]).{12,}$";
        Pattern pattern = Pattern.compile(regex);
        Matcher matcher = pattern.matcher(password);
        return matcher.matches();
    }
}
```



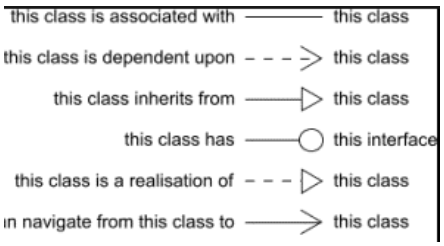

```java
    public static void main(String[] args) {
        //"Hello World"
        String string = new String(original:"Hello");
        String string2 = string.concat(str:"_World");
        System.out.println(string2);
        String string3 = string2.replace(target:"_", replacement:" ");
        System.out.println(string3);

        StringBuffer buffer = new StringBuffer(str:"Hello");
        buffer.append(str:"_World");
        System.out.println(buffer);
        buffer.setCharAt(index:5, ch:' ');
        System.out.println(buffer);
    }
```

```java
//IEEEFormatStrategyTest.java
public class IEEEFormatStrategyTest {
    @Test
    public void testFormatName(){
        CitationStrategy strategy = new IEEEFormatStrategy();
        assertEquals("B. Esfandiari", strategy.formatName("Babak", "Esfandiari"));
    }
}
```

```java
// TodoController.java
public class TodoController {
    private TodoComposite todoList;
    private TodoView view;

    public TodoController(TodoComposite todoList, TodoView view) {
        this.todoList = todoList;
        this.view = view;
    }

    public void saveTodoListToFile(String filename) {
        try (ObjectOutputStream out = new ObjectOutputStream(new FileOutputStream(filename))) {
            out.writeObject(todoList);
        } catch (IOException e) {
            e.printStackTrace();
        }
    }

    public void loadTodoListFromFile(String filename) {
        try (ObjectInputStream in = new ObjectInputStream(new FileInputStream(filename))) {
            todoList = (TodoComposite) in.readObject();
        } catch (IOException | ClassNotFoundException e) {
            e.printStackTrace();
        }
    }

    public void addTodoItem(TodoComposite item) {
        todoList.add(item);
    }

    public void displayTodoList() {
        view.displayTodoList(todoList);
    }    You, 17 hours ago • Adding SYSC3110 study material

    public void displayXMLTodoList() {
        view.displayXMLTodoList(todoList);
    }
}
```

```java
// Main.java
public class Main {
    Run | Debug
    public static void main(String[] args) {
        TodoProject mainProject = new TodoProject(name:"Main Project");
        TodoItem item1 = new TodoItem(text:"Complete Java assignment");
        TodoItem item2 = new TodoItem(text:"Read design patterns book");

        mainProject.add(item1);
        mainProject.add(item2);

        TodoProject subProject = new TodoProject(name:"Sub Project");
        subProject.add(new TodoItem(text:"Sub task 1"));
        subProject.add(new TodoItem(text:"Sub task 2"));

        mainProject.add(subProject);

        TodoView view = new TodoView();
        TodoController controller = new TodoController(mainProject, view);

        controller.displayTodoList();
        controller.displayXMLTodoList();

        controller.saveTodoListToFile(filename:"todolist.ser");
        controller.loadTodoListFromFile(filename:"todolist.ser");

    }
}
```

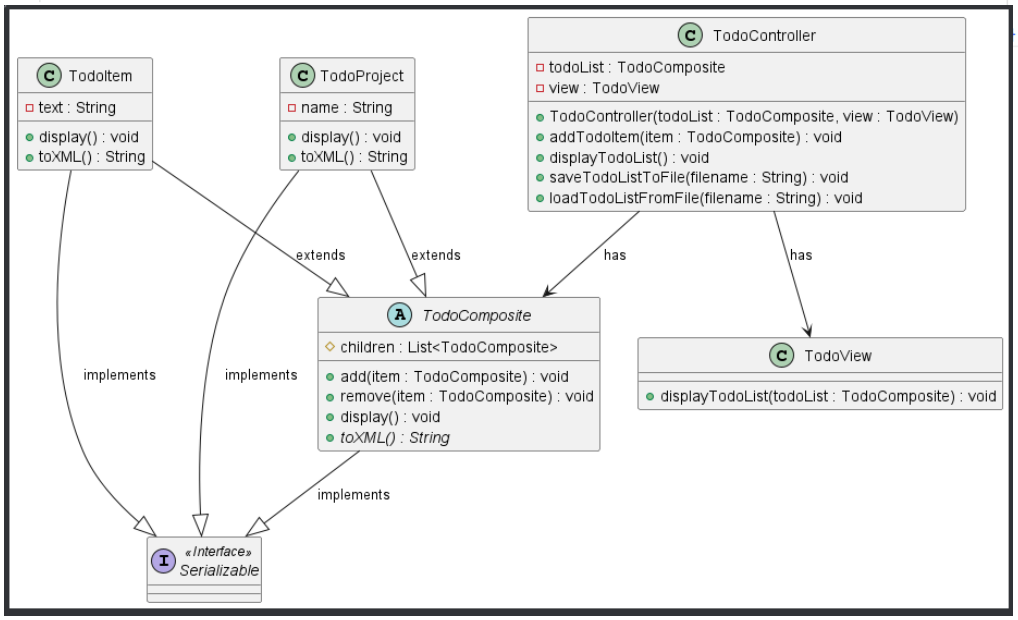
```java
//TodoProject.java
public class TodoProject extends TodoComposite {
    private String name;

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

    @Override
    public void display() {
        System.out.println(name + ":");
        for (TodoComposite item : children) {
            item.display();
        }
    }

    public String toXML() {
        StringBuilder xmlBuilder = new StringBuilder(str:"<todoProject>\n");
        for (TodoComposite item : children) {
            xmlBuilder.append(item.toXML()).append(str:"\n");
        }
        xmlBuilder.append(str:"</todoProject>");
        System.out.println(xmlBuilder.toString());
        return xmlBuilder.toString();
    }
```




```java
// TodoView.java
public class TodoView {
    public void displayTodoList(TodoComposite todoList) {
        todoList.display();
    }

    public void displayXMLTodoList(TodoComposite todoList) {
        todoList.toXML();
    }
}
```

```java
// TodoComposite.java
public abstract class TodoComposite implements Serializable {
    protected List<TodoComposite> children = new ArrayList<>();

    public void add(TodoComposite item) {
        children.add(item);
    }

    public void remove(TodoComposite item) {
        children.remove(item);
    }

    public abstract void display();

    public abstract String toXML();
}
```

MVC Todolist
Composite, XML,
Serialization




```java
//TodoItem.java
public class TodoItem extends TodoComposite {
    private String text;

    public TodoItem(String text) {
        this.text = text;
    }

    @Override
    public void display() {
        System.out.println("- " + text);
    }

    public String toXML() {
        return "<todo>" + this.text + "</todo>";
    }
}
```