



☐ +
 ☒ ▼
 ☐ ✕

☐ +
 ☒ ▼
 ☐ ✕

☐ +
 ☒ ▼
 ☐ ✕

☐ +
 ☒ ▼
 ☐ ✕

☐ +
 ☒ ▼
 ☐ ✕

☐ +
 ☒ ▼
 ☐ ✕

☐ +
 ☒ ▼
 ☐ ✕

☐ +
 ☒ ▼
 ☐ ✕

☐ +
 ☒ ▼
 ☐ ✕

☐ +
 ☒ ▼
 ☐ ✕

☐ +
 ☒ ▼
 ☐ ✕

☐ +
 ☒ ▼
 ☐ ✕

☐ +
 ☒ ▼
 ☐ ✕

☐ +
 ☒ ▼
 ☐ ✕

☐ +
 ☒ ▼
 ☐ ✕

☐ +
 ☒ ▼
 ☐ ✕

☐ +
 ☒ ▼
 ☐ ✕

☐ +
 ☒ ▼
 ☐ ✕

☐ +
 ☒ ▼
 ☐ ✕

☐ +
 ☒ ▼
 ☐ ✕

☐ +
 ☒ ▼
 ☐ ✕

☐ +
 ☒ ▼
 ☐ ✕

☐ +
 ☒ ▼
 ☐ ✕



=> Structural Design Patterns :-

- > These patterns deal with the composition of classes and objects to form larger structures.
- > They facilitate the creation of relationships between objects to build more complex systems.
- > There are 7 types of Structural Design Patterns :-

1. Adapter Design Pattern:

= Allows incompatible interfaces to work together.

2. Bridge Design Pattern:

= Separates an object's abstraction from its implementation.

3. Composite Design Pattern:

= Composes objects into tree structures to treat individual objects and compositions uniformly.

4. Decorator Design Pattern:

= Dynamically adds behavior to objects without altering their class.

5. Facade Design Pattern:

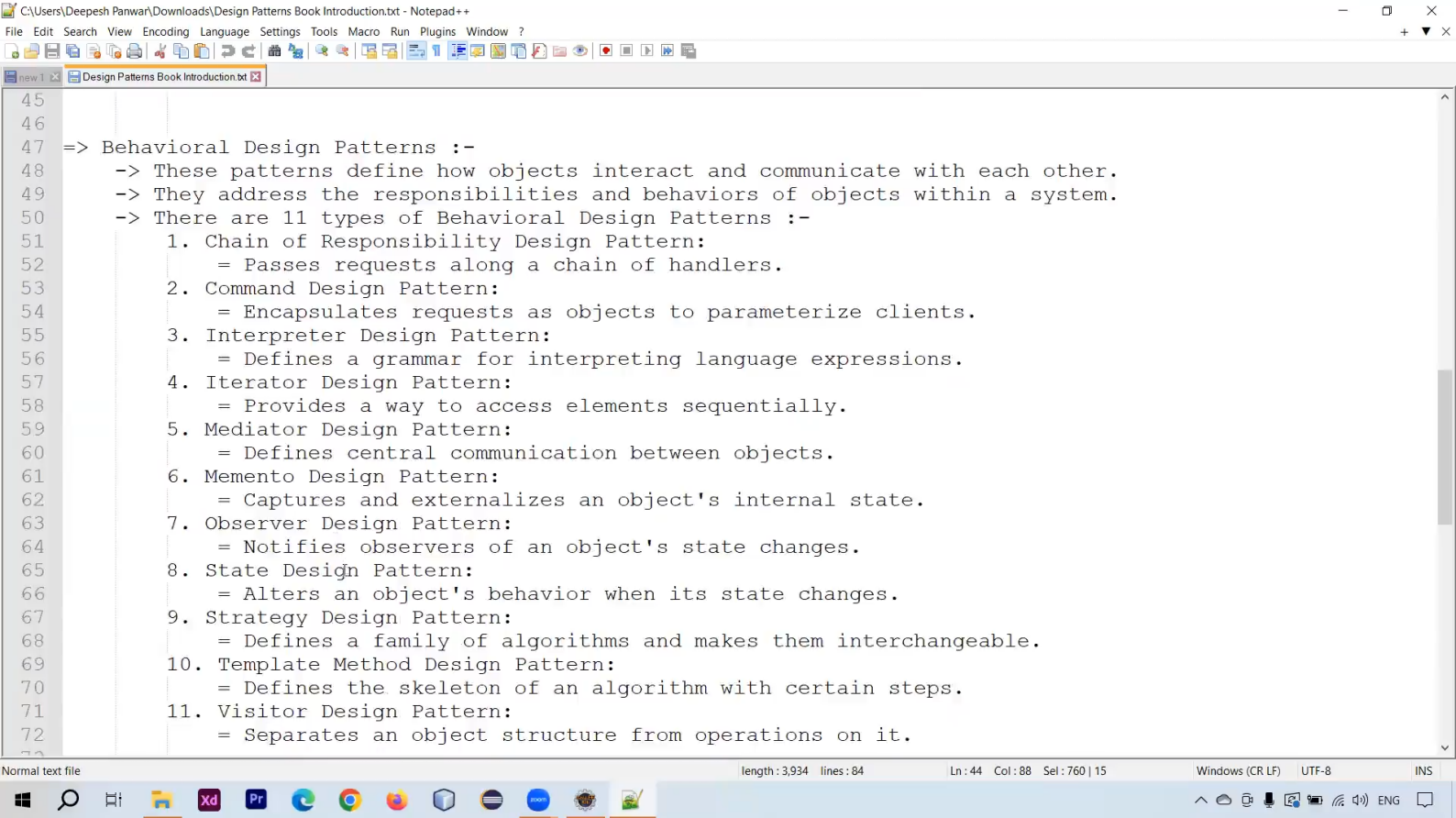
= Provides a simplified interface to a complex subsystem.

6. Flyweight Design Pattern:

= Shares common, stateless objects to conserve memory.

7. Proxy Design Pattern:

= Provides a surrogate or placeholder for another object to control access.



==> Miscellaneous Design Patterns :-

-> These are not defined in the above mentioned book

-> Some examples are :-

1. Dependency Injection Design Pattern

2. DAO Design Pattern

3. MVC Design Pattern

etc