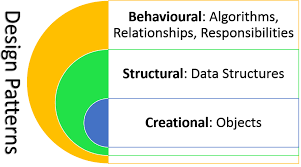
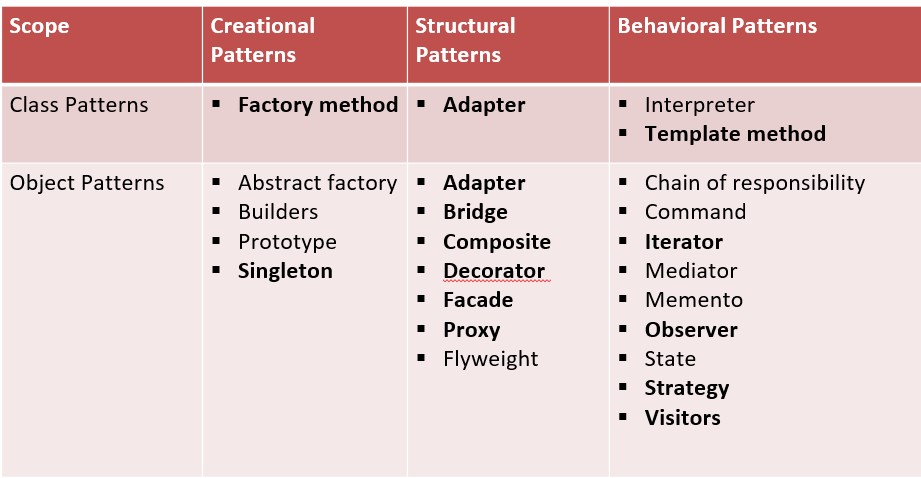
**Design Pattern**

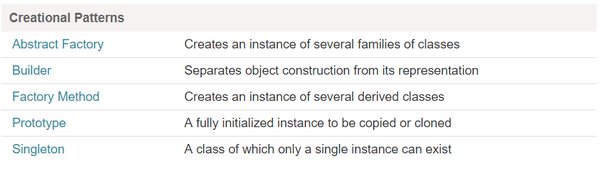
1. **Creational Patterns:** These design patterns provide ways to create objects while hiding the *creation logic*, instead of instantiating objects directly using the new operator. This gives the program more flexibility in deciding which objects need to be created for a given *use case*.
2. **Structural Patterns:** These design patterns deal with class and object composition. The concept of inheritance is used to compose interfaces and define ways to compose objects to obtain new functionality.
3. **Behavioral Patterns:** These design patterns are specifically concerned with communication between objects.

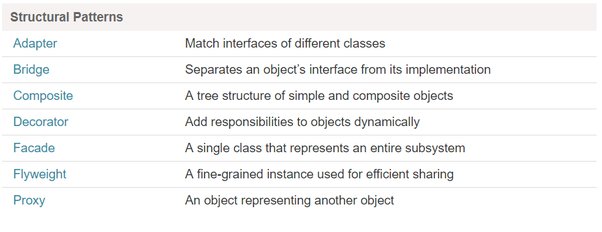
**1) Strategy Design pattern:**

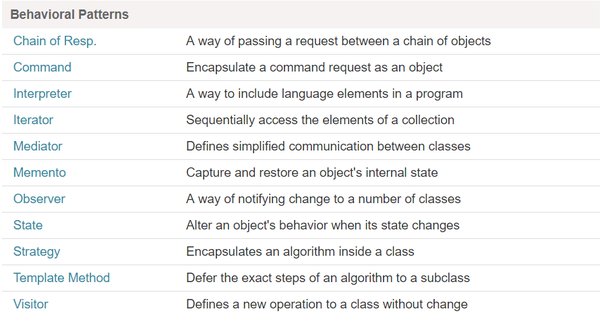
**Multiple algorithm to perform a particular task. Which algorithm is right it decide at runtime.**

**Basically encapsulate different behaviour of strategy into separate classes and dyamic decided.**

****







**Creation**

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |

**Structure**

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |

**Behaviour**

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |