**Project Proposal**

Design patterns represent the best practices used by experienced object-oriented software developers. Design patterns are solutions to general problems that software developers faced during software development. These solutions were obtained by trial and error by numerous software developers over quite a substantial period of time. Java design patterns are classified in three categories: Structural, Behavioral and Creational patterns.

Structural design patterns are [design patterns](https://en.wikipedia.org/wiki/Design_pattern_(computer_science)) that ease the design by identifying a simple way to realize relationships between entities. Examples of structural design patterns are bridge pattern, adapter pattern, composite pattern etc.

Behavioral design patterns are concerned with the interaction and responsibility of objects. In these design patterns, the interaction between the objects should be in such a way that they can easily talk to each other and still should be loosely coupled. In the behavioral design pattern the implementation and the client should be loosely coupled in order to avoid hard coding and dependencies. Examples of behavioral design patterns are Iterator Pattern, Mediator Pattern, Memento Pattern, Observer Pattern etc.

Creational design patterns are [design patterns](https://en.wikipedia.org/wiki/Design_pattern_(computer_science)) that deal with [object creation](https://en.wikipedia.org/wiki/Object_lifetime) mechanisms, trying to create objects in a manner suitable to the situation. The basic form of object creation could result in design problems or in added complexity to the design. Creational design patterns solve this problem by somehow controlling this object creation.

I will use “**Factory Method design pattern**” in java for this research project. Factory method pattern is one of the most used design patterns in Java. This type of design pattern comes under “**creational pattern”** as this pattern provides one of the best ways to create an object. Project will introduce background and advantages of Factory design pattern. I will implement this using a small java code and will explain with class diagrams and verify with the output of the project code.

Factory pattern is most suitable when there is some complex object creation steps are involved. To ensure that these steps are centralized and not exposed to composing classes, factory pattern should be used. The Factory Method Pattern is also known as **Virtual** **Constructor**.

The Advantages of using Factory Method Design Pattern are:

* Factory Method Pattern allows the sub-classes to choose the type of objects to create.
* It promotes the loose-coupling by eliminating the need to bind application-specific classes into the code

#### The Factory Method design pattern usage:

* When a class doesn't know what sub-classes will be required to create
* When a class wants that its sub-classes specify the objects to be created.
* When the parent classes choose the creation of objects to its sub-classes.