## Creational Design Patterns

## Overview

* Provides a way to instantiate objects efficiently

## Singleton

* Private constructor to restrict instantiation of the class from other classes.
* Private static variable of the same class that is the only instance of the class
* Public static method that returns the instance of the class, this is the global access point for outer world to get the instance of the singleton class

### Factory

* Factory design pattern is used when we have a super class with multiple sub-classes and based on input, we need to return one of the sub-class.

### Abstract Factory

* Similar to Factory, but it is factory of factories

### Builder

* Provides a way to instantiate objects efficiently

### Prototype

* Provides a way to instantiate objects efficiently