

## Lab 2 Q3

- A. You can use a factory Method pattern when a class can't anticipate the class of objects it must create. You can also use it when a class wants it's subclasses to specify the objects it creates. Classes delegate responsibility to one of several helper subclasses, and you want to localize the knowledge of which helper subclasses is the delegate.
- B. **Product (NumberList)** – Abstract class which defines all of the lists which will be created
- ConcreteProduct(IntList, DoubleList, HexList)**- This the implementation of NumberList and each subclass represents a different type of number list.
- Creator(NumberFactory)**- This is the factory class which decides which NumberList subclass to create. It uses functions to determine the correct concrete product based on the input.
- Client(NumberFactoryTest)**-This is the class that uses the factory to get number list objects. This class doesn't directly instantiate the Concrete Product like IntList instead it asks NumberFactory to provide a NumberList object.
- C. **Prouduct-** NumberList
- Concrete Product-** IntList, DoubleList, HexList
- Creator-** NumberFactory
- Client-** NumberFactoryTest