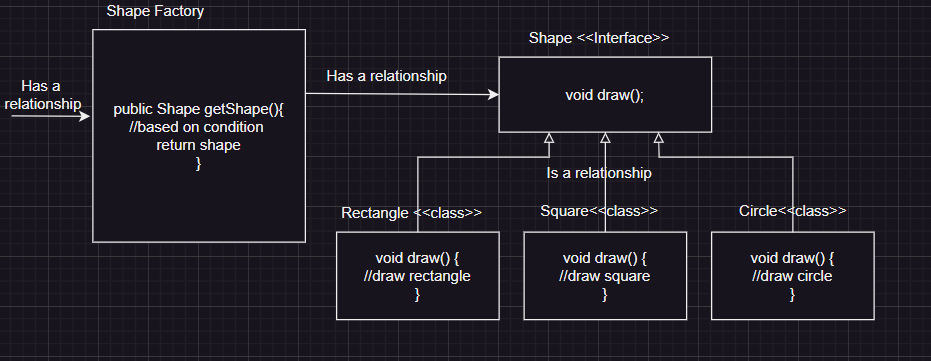
**Factory Pattern:** The factory design pattern is used when we have a superclass with multiple sub-classes and based on input, we need to return one of the sub-class. This pattern takes out the responsibility of the instantiation of a class from the client program to the factory class. Let’s first learn how to implement a factory design pattern in java and then we will look into factory pattern advantages. We will see some of the factory design pattern usage in JDK. Note that this pattern is also known as **Factory Method Design Pattern**.



**When to Use the Factory Pattern?**

* When the exact class of the object is not known until runtime.
* When you want to abstract the instantiation process and make it more flexible or reusable.
* To promote loose coupling between the client code and the concrete classes.

[**Factory Design Pattern Advantages**](https://www.digitalocean.com/community/tutorials/factory-design-pattern-in-java#factory-design-pattern-advantages)

1. Factory design pattern provides approach to code for interface rather than implementation.
2. Factory pattern removes the instantiation of actual implementation classes from client code. Factory pattern makes our code more robust, less coupled and easy to extend. For example, we can easily change PC class implementation because client program is unaware of this.
3. Factory pattern provides abstraction between implementation and client classes through inheritance.