

Simple Factory

Suppose you are designing a GUI Button for Windows which can have several implementations; RoundedButton, SquareButton, ...

Refactor the following client code and implement all necessary classes in order to use Simple Factory

```
public class ButtonDesign {

    public Button drawButton(String type) {
        Button button;

        if(type.equals("rounded")){
            button = new RoundedButton();
        }else if(type.equals("square")){
            button = new SquareButton();
        }

        return button;
    }
    //process on the created button
    Button.draw();
}
```

Factory Method

Suppose we want to expand the design to allow drawing several implementations. a GUI Button which can have several implementations based on the OS; MacButton which has an implementation suitable for Mac devices, WindowsButton suitable for Windows machines

Abstract Factory

We want to make use of the abstract factory pattern, to enable creating UI themes which consist of Buttons and CheckBoxes. The factory (GUIFactory) is the abstract factory that can create two abstract products; Button and CheckBox. For each OS, we can have a concrete GUI factory. For now, let's assume that we need one for Mac and one for Windows

- a) Draw the class diagram of the main classes involved in the design
- b) Write code