

}

### Client.java

```
package com.saket.demo.proxy;

public class Client
{
    public static void main (String[] args)
    {
        Internet internet = new ProxyInternet();
        try
        {
            internet.connectTo("geeksforgeeks.org");
            internet.connectTo("abc.com");
        }
        catch (Exception e)
        {
            System.out.println(e.getMessage());
        }
    }
}
```

As one of the site is mentioned in the banned sites, So  
Running the program will give the output :

```
Connecting to geeksforgeeks.org
Access Denied
```

### Benefits:

- One of the advantages of Proxy pattern is security.
- This pattern avoids duplication of objects which might be huge size and memory intensive. This in turn increases the performance of the application.
- The remote proxy also ensures about security by installing the local code proxy (stub) in the client machine and then accessing the server with help of the remote code.

### Drawbacks/Consequences:

This pattern introduces another layer of abstraction which sometimes may be an issue if the RealSubject code is accessed by some of the clients directly and some of them might access the Proxy classes. This might cause disparate behaviour.

### Interesting points:

- There are few differences between the related patterns. Like Adapter pattern gives a different interface to its subject, while Proxy patterns provides the same interface from the original object but the decorator provides an enhanced interface. Decorator pattern adds additional behaviour at runtime.
- Proxy used in Java API: java.rmi.\*;

