



```
public class CreditCardTerminal {  
  
    public CreditCardTerminal(){};  
  
    public processCard(PaymentProcessor pp, Card c){  
        pp.processCard(c);  
    }  
}  
  
public class PaymentProcessor{  
  
    public PaymentProcessor(){  
        initalize();  
    }  
  
    public void processCard (Card c){  
        // Processing payment action ....  
    }  
  
    public void initalize(){  
        // Initalization things ....  
    }  
}
```


What is a Global Dependency?

- In programming a **dependency** is an object passed into a function or class in order to enable/provide functionality of that method or class.
 - Often this is done by passing the necessary object as a reference, then using the object's methods to complete the task.
 - In UML class diagrams dependencies are marked as dotted arrows from class A (the supplier) to
- A **global dependency** is a variable and/or class that is declared at a global scope and shared and accessed throughout the program
- **Global dependencies** are particularly hard to break free from as they are often **tightly coupled** to large parts of the programs and their usages are scattered throughout.

```
public class CreditCardTerminal {  
  
    public CreditCardTerminal(){};  
  
    public processCard(PaymentProcessor pp, Card c){  
        pp.processCard(c);  
    }  
}  
  
public class PaymentProcessor{  
  
    public PaymentProcessor(){  
        initialize();  
    }  
  
    public void processCard (Card c){  
        // Processing payment action ....  
    }  
  
    public void initialize(){  
        // Initialization things ....  
    }  
}
```

The Singleton