

# Mapping to Swift

## Basic Structure

```
enum UserRole {
    case admin, user, guest
}

enum ResourceType {
    case publicResource, sharedResource, privateResource
}

enum Action {
    case read, write, delete
}

enum AccessResult {
    case allow, deny

    var symbol: String { ... }
}

// Complete and exhaustive implementation
func checkAccess(
    role: UserRole,
    resource: ResourceType,
    action: Action,
    isOwner: Bool
) -> AccessResult {
    switch (role, resource, action, isOwner) { ... }
}
```

# Mapping to Swift Pattern Matching

1:1 mapping from table to  
code

```
switch (role, resource, action, isOwner) {  
    // Admin can do anything  
    case (.admin, _, _, _):  
        return .allow  
  
    // User access to public resources  
    case (.user, .publicResource, .read, _):  
        return .allow  
  
    // User access to shared resources  
    case (.user, .sharedResource, .read, _):  
        return .allow  
  
    // User access to owned resources (full control)  
    case (.user, _, .read, true):  
        return .allow  
    case (.user, _, .write, true):  
        return .allow  
    case (.user, _, .delete, true):  
        return .allow  
  
    // Guest access  
    case (.guest, .publicResource, .read, _):  
        return .allow  
  
    default:  
        return .deny
```