DCL01-PL. Do not reuse variable names in subscopes

Do not use the same variable name in two scopes where one scope is contained in another. For example,

- A lexical variable should not share the name of a package variable if the lexical variable is in a subscope of the global variable.
- A block should not declare a lexical variable with the same name as a lexical variable declared in any block that contains it.

Reusing variable names leads to programmer confusion about which variable is being modified. Additionally, if variable names are reused, generally one or both of the variable names are too generic.

Noncompliant Code Example

This noncompliant code example declares the $errmsg identifier at file scope and reuses the same identifier to declare a string that is private in the report_error() subroutine. Consequently, the program prints "The error is Local error" rather than "The error is Global error."

```
$_error(){$errmsg = "Global error";
sub report_error {
    my $errmsg = shift(@_);
    print "The error is $errmsg\n";
};
report_error("Local error");
```

Compliant Solution

This compliant solution uses different, more descriptive variable names.

```
$global_error_msg = "Global error";
sub report_error {
    my $local_error_msg = shift(@_);
    print "The error is $local_error_msg\n";
};
report_error("Local error");
```

When the block is small, the danger of reusing variable names is mitigated by the visibility of the immediate declaration. Even in this case, however, variable name reuse is not desirable. In general, the larger the declarative region of an identifier, the more descriptive and verbose should be the name of the identifier.

By using different variable names globally and locally, the compiler forces the developer to be more precise and descriptive with variable names.

Risk Assessment

Hiding variables in enclosing scopes can lead to surprising results.

<table>
<thead>
<tr>
<th>Recommendation</th>
<th>Severity</th>
<th>Likelihood</th>
<th>Remediation Cost</th>
<th>Priority</th>
<th>Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>DCL01-PL</td>
<td>Low</td>
<td>Probable</td>
<td>Medium</td>
<td>P4</td>
<td>L3</td>
</tr>
</tbody>
</table>

Automated Detection

<table>
<thead>
<tr>
<th>Tool</th>
<th>Version</th>
<th>Checker</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>B::Lint</td>
<td>5.0</td>
<td>.* masks earlier declaration in same scope</td>
<td>Implemented</td>
</tr>
</tbody>
</table>

Related Guidelines

- SEI CERT C Coding Standard
- DCL01-C. Do not reuse variable names in subscopes
- SEI CERT C++ Coding Standard
Bibliography

[Wall 2011] perlsub