Example from Tisdall, James, *Beginning Perl for Bioinformatics*, O'Reilly, 2001

```
#!/usr/bin/perl -w

# Example 4-4    Calculating the reverse complement of a strand of DNA

# The DNA
$DNA = 'ACGGGAGGACGGGAAAAATTACTACCGGCATTAGC';

# Print the DNA onto the screen
print "Here is the starting DNA:\n\n";
print "$DNA\n\n";

# Calculate the reverse complement
# Warning: this attempt will fail!
#
# First, copy the DNA into new variable $revcom
# (short for REVerse COMplement)
# Notice that variable names can use lowercase letters like
# "revcom" as well as uppercase like "DNA". In fact,
# lowercase is more common.
#
# It doesn't matter if we first reverse the string and then
# do the complementation; or if we first do the complementation
# and then reverse the string. Same result each time.
# So when we make the copy we'll do the reverse in the same statement.
#
$revcom = reverse $DNA;

#
# Next substitute all bases by their complements,
# A->T, T->A, G->C, C->G
#
$revcom =~ s/A/T/g;
$revcom =~ s/T/A/g;
$revcom =~ s/G/C/g;
$revcom =~ s/C/G/g;

# Print the reverse complement DNA onto the screen
print "Here is the reverse complement DNA:\n\n";
print "$revcom\n\n";

exit;
```
#!/usr/bin/perl -w
# Continues previous example - but does it right

# The DNA
$DNA = 'ACGGGAGGACGGAATTAATACGGGCATTAGC';

# Print the DNA onto the screen
print "Here is the starting DNA:

";
print "$DNA

";

# Make a new copy of the DNA
$revcom = reverse $DNA;

# Use tr/// instead of s///
$revcom =~ tr/ACGTacgt/TGCAtgca/;

# Print the reverse complement DNA onto the screen
print "Here is the reverse complement DNA:

";
print "$revcom

";
print "This time it worked!

";
exit;