PERL BITWISE OPERATORS EXAMPLE

There are following Bitwise operators supported by Perl language, assume if $a = 60; and b = 13$ −

Operator	Description	Example
&	Binary AND Operator copies a bit to the result if it exists in both operands.	$a & $b will give 12 which is 0000 1100
|	Binary OR Operator copies a bit if it exists in eather operand.	$a\|$b will give 61 which is 0011 1101
^	Binary XOR Operator copies the bit if it is set in one operand but not both.	$a^$b will give 49 which is 0011 0001
~	Binary Ones Complement Operator is unary and has the efect of 'flipping' bits.	$a will give -61 which is 1100 0011 in 2's complement form due to a signed binary number.
<<	Binary Left Shift Operator. The left operands value is moved left by the number of bits specified by the right operand.	$a << 2 will give 240 which is 1111 0000
>>	Binary Right Shift Operator. The left operands value is moved right by the number of bits specified by the right operand.	$a >> 2 will give 15 which is 0000 1111

Example

Try the following example to understand all the bitwise operators available in Perl. Copy and paste the following Perl program in test.pl file and execute this program.

```perl
#!/usr/local/bin/perl

use integer;

$a = 60;
$b = 13;

print "Value of \$a = $a and value of \$b = $b\n";

$c = $a & $b;
print "Value of \$a & \$b = $c\n";

$c = $a | $b;
print "Value of \$a | \$b = $c\n";

$c = $a ^ $b;
print "Value of \$a ^ \$b = $c\n";

$c = ~$a;
print "Value of ~\$a = $c\n";

$c = $a << 2;
print "Value of \$a << 2 = $c\n";

$c = $a >> 2;
print "Value of \$a >> 2 = $c\n";
```

When the above code is executed, it produces the following result −

```
Value of $a = 60 and value of $b = 13
Value of $a & $b = 12
Value of $a | $b = 61
Value of $a ^ $b = 49
Value of ~$a = 18446744073709551555
Value of $a << 2 = 240
Value of $a >> 2 = 15
```

```
Value of $a = 60 and value of $b = 13
Value of $a & $b = 12
Value of $a | $b = 61
Value of $a ^ $b = 49
Value of ~$a = 18446744073709551555
Value of $a << 2 = 240
Value of $a >> 2 = 15
```