ASSIGNMENT OPERATORS IN OBJECTIVE-C

There are following assignment operators supported by Objective-C language:

Operator	Description	Example
=	Simple assignment operator, Assigns values from right side operands to left side operand	C = A + B will assign value of A + B into C
+=	Add AND assignment operator, It adds right operand to the left operand and assigns the result to left operand	C += A is equivalent to C = C + A
-=	Subtract AND assignment operator, It subtracts right operand from the left operand and assigns the result to left operand	C -= A is equivalent to C = C - A
*=	Multiply AND assignment operator, It multiplies right operand with the left operand and assigns the result to left operand	C *= A is equivalent to C = C * A
/=	Divide AND assignment operator, It divides left operand with the right operand and assigns the result to left operand	C /= A is equivalent to C = C / A
%=	Modulus AND assignment operator, It takes modulus using two operands and assigns the result to left operand	C %= A is equivalent to C = C % A
<<=	Left shift AND assignment operator	C <<= 2 is same as C = C << 2
>>=	Right shift AND assignment operator	C >>= 2 is same as C = C >> 2
&=	Bitwise AND assignment operator	C &= 2 is same as C = C & 2
^=	bitwise exclusive OR and assignment operator	C ^= 2 is same as C = C ^ 2
|=	bitwise inclusive OR and assignment operator	C |= 2 is same as C = C | 2

Example

Try the following example to understand all the assignment operators available in Objective-C programming language:

```
#import <Foundation/Foundation.h>

main()
{
   int a = 21;
   int c ;

   c =  a;
   NSLog(@"Line 1 - =  Operator Example, Value of c = %d\n", c );

   c +=  a;
   NSLog(@"Line 2 - += Operator Example, Value of c = %d\n", c );

   c -=  a;
   NSLog(@"Line 3 - -= Operator Example, Value of c = %d\n", c );
```

```
   c *=  a;
   NSLog(@"Line 4 - *= Operator Example, Value of c = %d\n", c );

   c /=  a;
   NSLog(@"Line 5 - /= Operator Example, Value of c = %d\n", c );

   c  = 200;
   c %=  a;
   NSLog(@"Line 6 - %= Operator Example, Value of c = %d\n", c );

   c <<=  2;
   NSLog(@"Line 7 - <<= Operator Example, Value of c = %d\n", c );

   c >>=  2;
   NSLog(@"Line 8 - >>= Operator Example, Value of c = %d\n", c );

   c &=  2;
   NSLog(@"Line 9 - &= Operator Example, Value of c = %d\n", c );

   c ^=  2;
   NSLog(@"Line 10 - ^= Operator Example, Value of c = %d\n", c );

   c |=  2;
   NSLog(@"Line 11 - |= Operator Example, Value of c = %d\n", c );

}
```

When you compile and execute the above program, it produces the following result:

```
2013-09-07 22:00:19.263 demo[21858] Line 1 - =  Operator Example, Value of c = 21
2013-09-07 22:00:19.263 demo[21858] Line 2 - += Operator Example, Value of c = 42
2013-09-07 22:00:19.263 demo[21858] Line 3 - -= Operator Example, Value of c = 21
2013-09-07 22:00:19.263 demo[21858] Line 4 - *= Operator Example, Value of c = 441
2013-09-07 22:00:19.263 demo[21858] Line 5 - /= Operator Example, Value of c = 21
2013-09-07 22:00:19.264 demo[21858] Line 6 - %= Operator Example, Value of c = 11
2013-09-07 22:00:19.264 demo[21858] Line 7 - <<= Operator Example, Value of c = 44
2013-09-07 22:00:19.264 demo[21858] Line 8 - >>= Operator Example, Value of c = 11
2013-09-07 22:00:19.264 demo[21858] Line 9 - &= Operator Example, Value of c = 2
2013-09-07 22:00:19.264 demo[21858] Line 10 - ^= Operator Example, Value of c = 0
2013-09-07 22:00:19.264 demo[21858] Line 11 - |= Operator Example, Value of c = 2
```