C Programlama

Öğretim Elemanı Bilgileri

- Dr. Öğr. Üyesi Sema ATASEVER
- Nevşehir Hacı Bektaş Veli Üniversitesi, Mühendislik Mim.Fak. Bilgisayar Mühendisliği
- Web sayfası : https://biz.nevsehir.edu.tr/sema/tr
- Email : sema@nevsehir.edu.tr | s.atasever@gmail.com

Ölçme Yöntemi

ARA SINAV

- Ara sınav : 100 puan üzerinden değerlendirilecektir , Katkı : %40

FİNAL ÖDEVİ

- Final Sınavı : 100 puan üzerinden değerlendirilecektir , Katkı : %60
- Nihai ders notu hesabı : Ara sınavın %40'ı, Final notunun %60'ı alınarak hesaplanmaktadır!

13. Hafta Konuları

- Bit operatörleri (AND, OR, XOR, sola kaydırma, sağa kaydırma, maskeleme), C'deki operatör kullanımları-özet, örnek kod uygulamaları.

Bit operatörleri

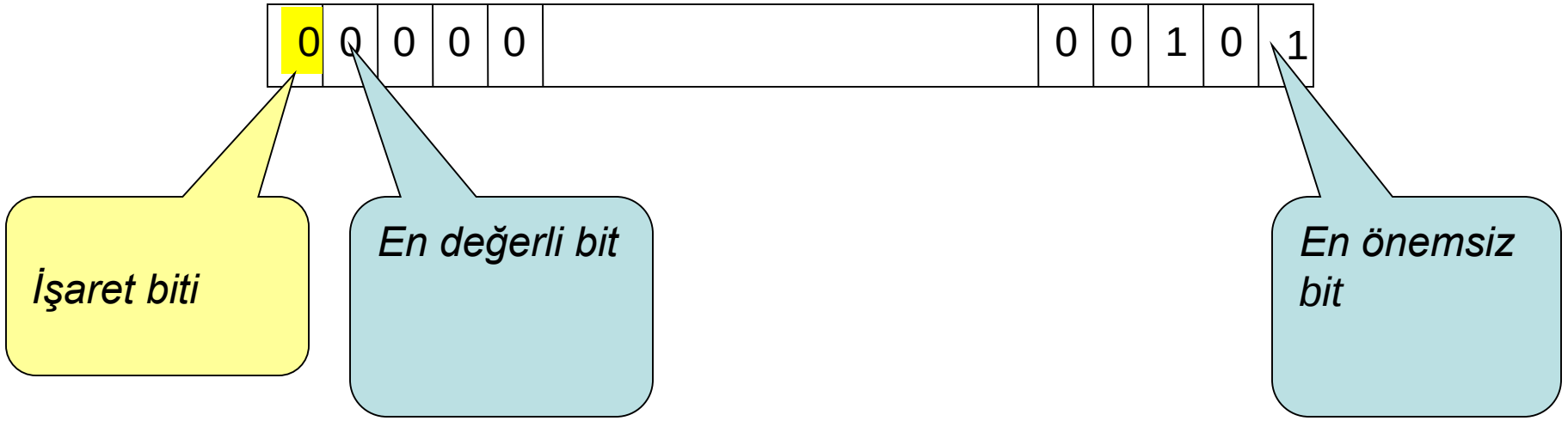
- Bir baytın en sağdaki biti, en az önemli veya düşük sıralı bit olarak bilinirken, en soldaki bit, en önemli veya yüksek sıralı bit olarak bilinir.
- Negatif sayıların temsili biraz farklı ele alınır. Çoğu bilgisayar bu tür sayıları 2'ye tümleyen (two's complement) notasyonu kullanarak temsil eder.
- Bu gösterimi kullanarak, en soldaki bit işaret bitini temsil eder. Bu bit 1 ise, sayı negatiftir; aksi takdirde, bit 0'dır ve sayı pozitiftir. Kalan bitler, sayının değerini temsil eder. İkiye tümleyen notasyonunda, –1 değeri tüm bitlerin 1'e eşit olmasıyla temsil edilir: 11111111

Bit operatörleri

- Negatif bir sayıyı ondalık sayıdan ikilik sayıya dönüştürmek için,
  - önce değere 1 eklenir,
  - sonucun mutlak değeri ikili olarak ifade edilir,
  - ardından tüm 0’lar 1’e, tüm 1’ler 0’a çevrilir.
  - Örneğin, -5'i ikilik sayı sisteminde ifade etmek için önce 1 eklenir ve sonuç -4 olur. İkili olarak ifade edilen 4 00000100'dır ve bitlerin tümleyeni 11111011'i üretir.

Tam sayıların ikili gösterimi

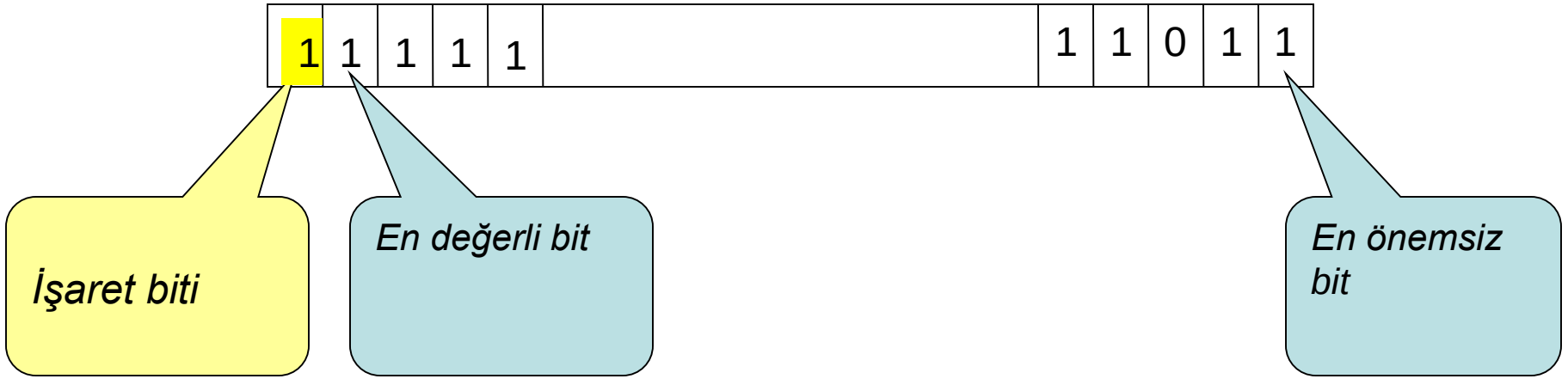
- Pozitif tam sayılar:



1*2^0+0*2^1+1*2^2=5

Tam sayıların ikili gösterimi

- Negatif tamsayılar



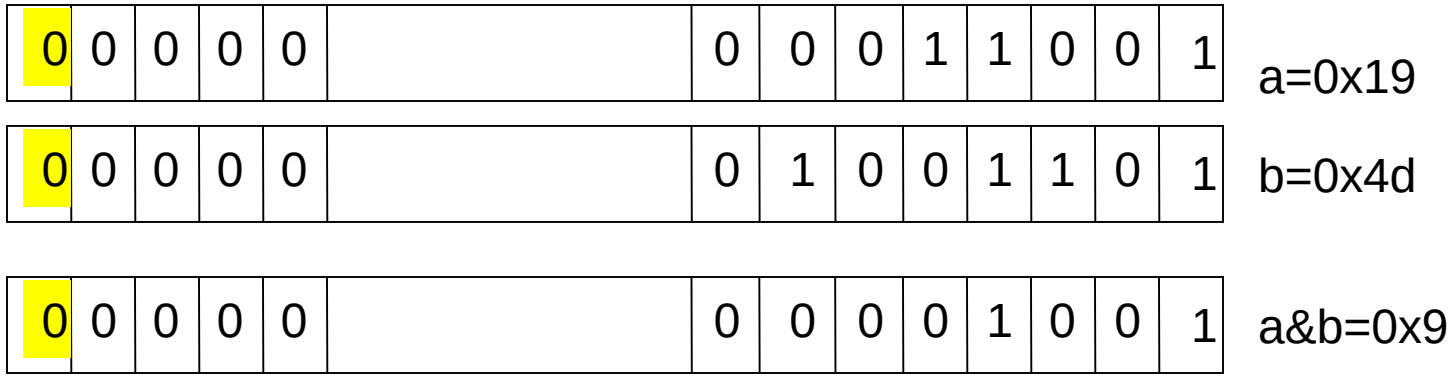
Bit operatörleri

- C dili bitler üzerinde işlem yapmak için özel olarak tasarlanmış bir dizi operatör sağlar.

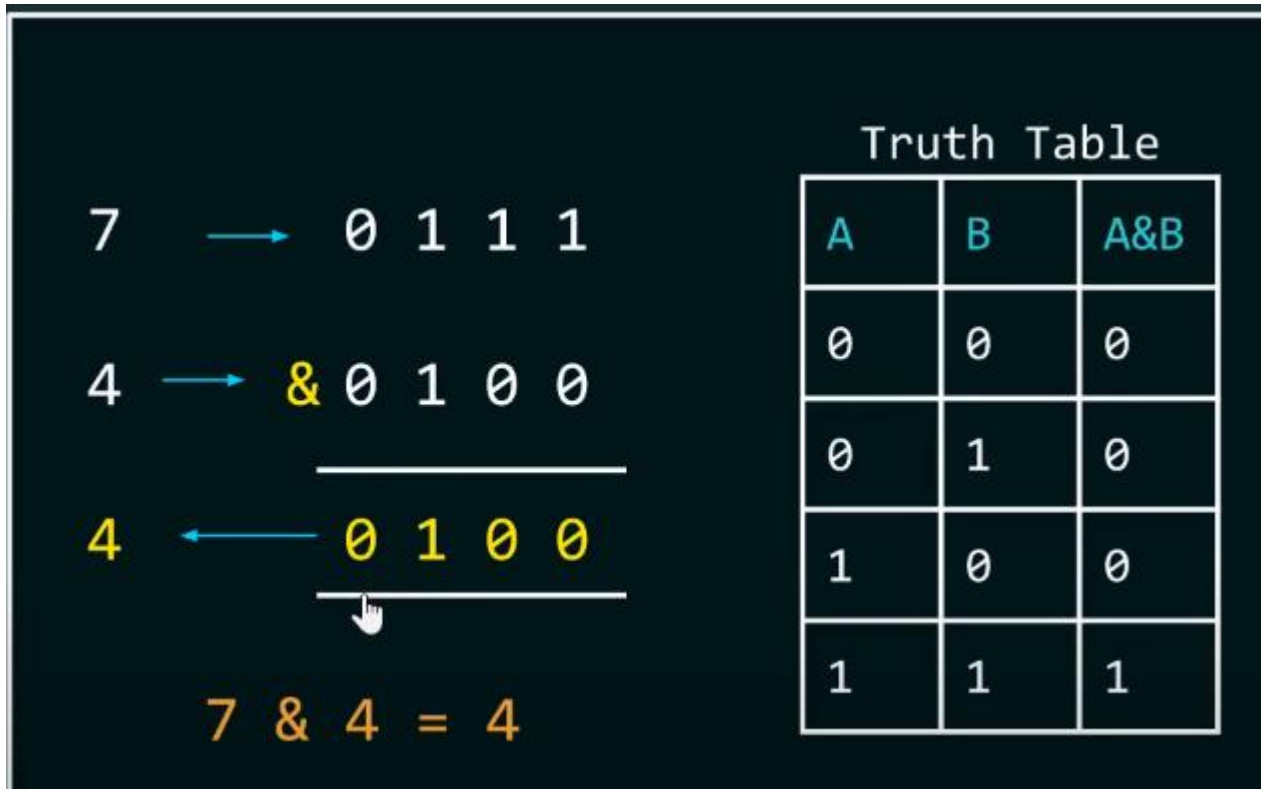
Bitsel(Bitwise) AND

b1	b2	b1&b2
0	0	0
0	1	0
1	0	0
1	1	1

```
int a=25;
int b=77;

printf("%x %x %x",a,b,a&b);
```



Bitsel(Bitwise) AND



https://www.youtube.com/watch?v=jlQmeyce65Q&t=6s

Örnek: tek / çift ?

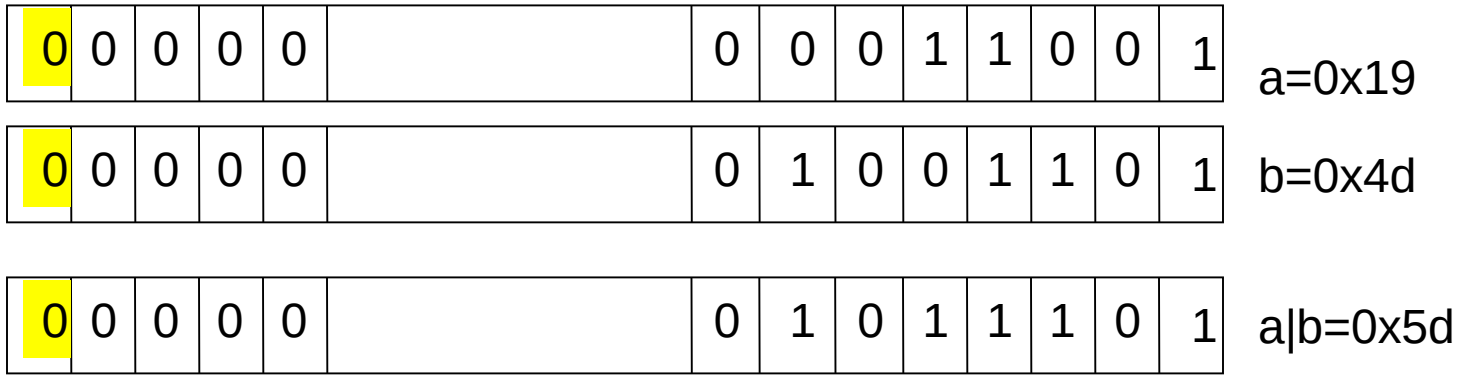
- Bitsel operatörleri kullanarak bir tamsayının çift mi yoksa tek mi olduğunu test etmek:
- Herhangi bir tek tamsayının en sağdaki biti 1'dir ve herhangi bir çift tamsayının en sağdaki biti 0'dır.

```
#include <stdio.h>
int main()
{
    int n=0;
    printf("Enter integer:");
    scanf("%d",&n);

    if ( n & 1 )
        printf("\nodd | %d", n & 1); // tek
    else
        printf("\neven | %d", n & 1); // çift

  getche();
}
```

Bitwise **Inclusive-OR (OR)**

b1	b2	b1|b2
0	0	0
0	1	1
1	0	1
1	1	1

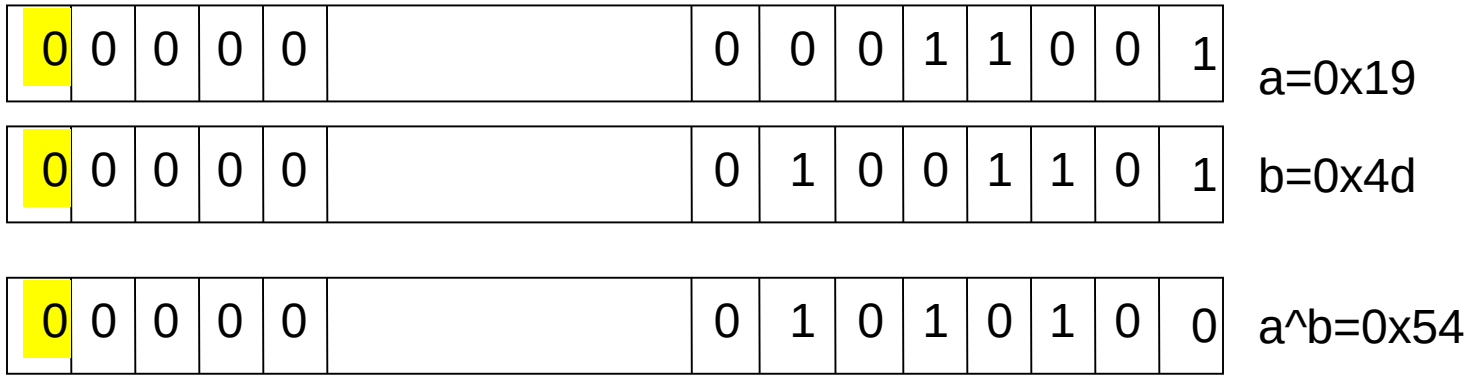
```
int a=25;
int b=77;

printf("%x %x %x",a,b,a|b);
```



Bitwise **Exclusive-OR (XOR)**

b1	b2	b1^b2
0	0	0
0	1	1
1	0	1
1	1	0

```
int a=25;
int b=77;

printf("%x %x %x",a,b,a^b);
```



Örnek : Bitwise **Exclusive-OR (XOR)**

```c
#include <stdio.h>
#include <stdlib.h>

_Bool oppositeSigns(int x,int y){
    printf("%d \n", x^y);
    return ( (x^y) <0 );
}

int main()
{
    int x=4, y=-2;

    if ( oppositeSigns(x,y) )
        printf("Signs are opposite");
    else
        printf("Signs are not opposite");
    return 0;
}
```

Output

```
-6
Signs are opposite
```

https://www.geeksforgeeks.org/detect-if-two-integers-have-opposite-signs/

1’e Tümleyen (The Ones Complement Operator)

b1	~b1
0	1
1	0

```c
#include <stdio.h>
main() {
    int a=25;

    printf("Hex: %x %x | Dec: %d %d \n",a,~a,a,~a);
}
```

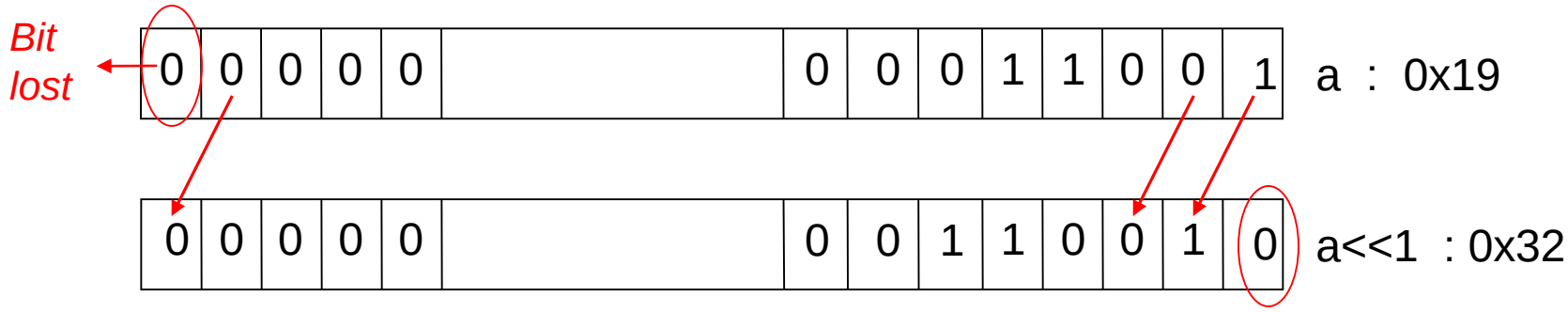
```
Hex: 19 ffffffe6 | Dec: 25 -26
```

0	0	0	0	0		0	0	0	1	1	0	0	1	a=0x19

1	1	1	1	1		1	1	1	0	0	1	1	0	~a=0xffffffe6

Sola Kaydırma Operatörü (The Left Shift Operator)

```
int a=25;

printf("%x %x",a,a<<1);
```



Örnek: 2 ile çarpma

- Bir değerin ikinin kuvvetiyle çarpılması: değeri uygun sayıda basamak sola kaydırmak ile mümkün olmaktadır.

```
int a;
scanf("%i",&a);
printf("a multiplied with 2 is %i \n",a<<1);
printf("a multiplied with 4 is %i \n",a<<2);
```

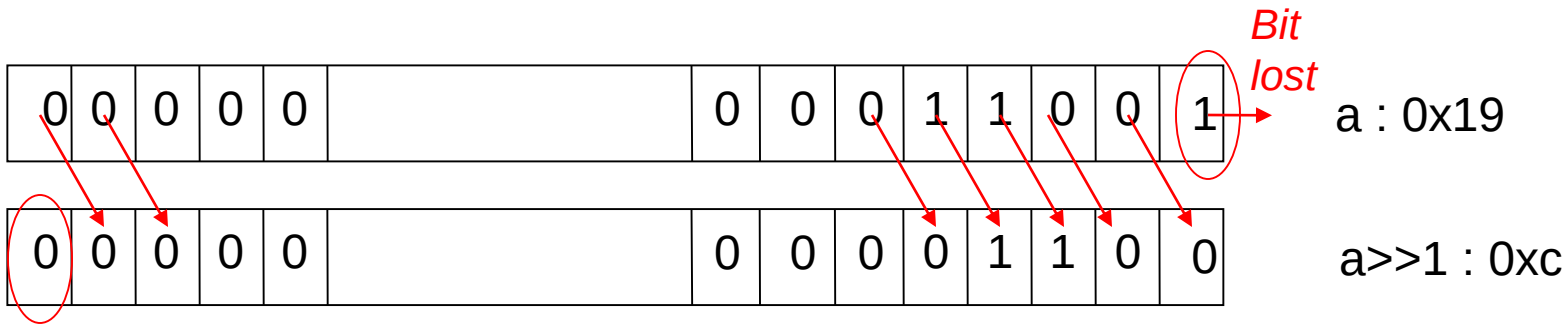
Output:

```
Enter the number:3
3 multiplied with 2 is 6
3 multiplied with 4 is 12
```

Sağa Kaydırma Operatörü (The Right Shift Operator)

- Bir değerin bitlerini sağa kaydırır.
- Değerin düşük sıralı bitinden dışarı kaydırılan bitler kaybolur.

```
int a=25;

printf("%x %x",a,a>>1);
```



Örnek: bit gruplarını elde etmek (extract groups of bits)

- Örnek: renk değerlerini temsil etmek için işaretsiz uzun bir değer kullanılır; düşük sıralı bayt kırmızı yoğunluğu, sonraki bayt yeşil yoğunluğu ve üçüncü bayt mavi yoğunluğu tutar.
- Daha sonra her rengin yoğunluğunu kendi işaretsiz karakter değişkeninde saklamak isterseniz yandaki kod parçasından yararlanabilirsiniz.

```c
#include <stdio.h>
#include <stdlib.h>

int main()
{
    unsigned char BYTE_MASK=0xff;
    unsigned long color=0x002a162f;
    unsigned char blue, green, red;

    red=color & BYTE_MASK;
    green=(color >>8) & BYTE_MASK;
    blue=(color >>16) & BYTE_MASK;
    printf("red=%x, green=%x, blue=%x", red, green, blue);
}
```

Özet: C'deki Operatörler

Table A.5 **Summary of C Operators**

Operator	Description	Associativity
`()`	Function call	
`[]`	Array element reference	
`->`	Pointer to structure member reference	Left to right
`.`	Structure member reference	
`-`	Unary minus	
`+`	Unary plus	
`++`	Increment	
`--`	Decrement	
`!`	Logical negation	
`~`	Ones complement	Right to left
`*`	Pointer reference (indirection)	
`&`	Address	
`sizeof`	Size of an object	
(*type*)	Type cast (conversion)	

*	Multiplication	
/	Division	Left to right
%	Modulus	
+	Addition	Left to right
-	Subtraction	
<<	Left shift	Left to right
>>	Right shift	
<	Less than	
<=	Less than or equal to	Left to right
>	Greater than	
=>	Greater than or equal to	

Table A.5 **Continued**

Operator	Description	Associativity
==	Equality	Left to right
!=	Inequality	
&	Bitwise AND	Left to right
^	Bitwise XOR	Left to right
|	Bitwise OR	Left to right
&&	Logical AND	Left to right
||	Logical OR	Left to right
?:	Conditional	Right to left
=		
*= /= %=		
+= -= &=	Assignment operators	Right to left
^= |=		
<<= >>=		
,	Comma operator	Right to left