

# **| Basis and Practice in Programming**

## week3

# Various Operators in C

- Substitution and arithmetic operator

Operator	Example	Meaning
=	a=20	Assign the value to variable a
+	a=4+3	Addition
-	a=4-3	Subtraction
*	a=4*3	Multiply
/	a=4/3	Division
%	a=4%3	Remainder

# Various Operators in C

- Increment and decrement operator

Operator	Example	Meaning
<code>++a</code>	<code>printf("%d", ++a)</code>	Increase first, operate later
<code>a++</code>	<code>printf("%d", a++)</code>	Operate first, increase later
<code>--b</code>	<code>printf("%d", --a)</code>	Decrease first, operate later
<code>b--</code>	<code>printf("%d", a--)</code>	Operate first, decrease later

# Various Operators in C

- Increment and decrement operator

```
/* week 3 practice 1 */
#include <stdio.h>

int main(void)
{
    int val1=10;
    int val2=10;

    printf("Operate first, increase later : %d \Wn", val1++);
    printf("Print again : %d \Wn\Wn", val1);

    printf("Increase first, operate later : %d \Wn", ++val2);
    printf(" Print again : %d \Wn", val2);

    return 0;
}
```

# Various Operators in C

- Increment and decrement operator

```
/* week 3 practice 2 */
#include <stdio.h>

int main(void)
{
    int val1=10;
    int val2=(val1--)+2;

    printf("val1 : %d \n", val1);
    printf("val2 : %d \n", val2);

    return 0;
}
```

```
/* week 3 practice 3 */
#include <stdio.h>

int main(void)
{
    int val1=10;
    int val2=(--val1)+2;

    printf("val1 : %d \n", val1);
    printf("val2 : %d \n", val2);

    return 0;
}
```

# Various Operators in C

- Association operators
  - Represents association between two operands
  - Returns true(1) or false(0)

Operator	Example	Meaning
<	a<b	Is a less than b?
>	a>b	Is a larger than b?
==	a==b	Is a equal to b?
!=	a!=b	Is a not equal to b?
<=	a<=b	Is a less than or equal to b?
>=	a>=b	Is a larger than or equal to b?

# Various Operators in C

- Association operators

```
/* week 3 practice 4 */
#include <stdio.h>

int main(void)
{
    int val1=10;
    int val2=12;
    int result1, result2, result3;

    result1=(val1==val2);
    result2=(val1<=val2);
    result3=(val1>val2);

    printf("result1 : %d \n", result1);
    printf("result2 : %d \n", result2);
    printf("result3 : %d \n", result3);

    return 0;
}
```

# Various Operators in C

- Logic operators
  - Represents and, or, not
  - Returns true(1) or false(0)

Operator	Example	Meaning
&&	a&&b	Returns true(1) when both a and b is true
	a  b	Returns true(1) when either a or b is true (also true when both a and b are true)
!	!a	Returns false when a is true and vice versa



# Various Operators in C

- Logic operators

```
/* week 3 practice 5 */
#include <stdio.h>

int main(void)
{
    int val1=10;
    int val2=12;
    int result1, result2, result3;

    result1=(val1==10 && val2==12);
    result2=(val1<12 || val2>12);
    result3=!val1;

    printf("result1 : %d \n", result1);
    printf("result2 : %d \n", result2);
    printf("result3 : %d \n", result3);

    return 0;
}
```

# Various Operators in C

- Bit manipulation operators

Operator	Example	Meaning
&	a&b	Binary AND Operator copies a bit to the result if it exists in both operands.
	a b	Binary OR Operator copies a bit if it exists in either operand.
^	a^b	Binary XOR Operator copies the bit if it is set in one operand but not both.
~	~a	Binary Ones Complement Operator is unary and has the effect of 'flipping' bits.
<<	a<<2	Binary Left Shift Operator. The left operands value is moved left by the number of bits specified by the right operand.
>>	a>>2	Binary Right Shift Operator. The left operands value is moved right by the number of bits specified by the right operand.

# Various Operators in C

- Bit manipulation operators
  - Example: and operator

Operand A	Operand B	Result
0	0	0
0	1	0
1	0	0
1	1	1

```
A :    0000 0101
B :    0000 0100
-----
Result: 0000 0100
```

# Various Operators in C

- Bit manipulation operators
  - Example: or operator

Operand A	Operand B	Result
0	0	0
0	1	1
1	0	1
1	1	1

```
A :    0000 0101
B :    0000 0100
-----
Result: 0000 0101
```

# Various Operators in C

- Bit manipulation operators
  - Example: xor operator

Operand A	Operand B	Result
0	0	0
0	1	1
1	0	1
1	1	0

```
A :    0000 0101
B :    0000 0100
-----
Result: 0000 0001
```

# Various Operators in C

- Bit manipulation operators
  - Example: binary one's complement operator

Operand A	Result
0	1
1	0

```
A :    0000 0101
-----
Result: 1111 1010
```

# Various Operators in C

- Bit manipulation operators
  - Example: shift operator

```
int a = 5;  
a = a << 1;
```

```
A :    0000 0101  
-----  
Result: 0000 1010
```

# Various Operators in C

- Bit manipulation operators
  - Week 3 exercise
    - Input two integer numbers A and B from user
    - Print the result of following bit operations
    - Due date : today 11:59 PM (Sep. 23. 2012)

