

Complex Selections

Instructor: Chien-Ho Ko

Outlines

- Introduction
- Complex **predicates**
- Complex selection **structures**
- *switch* and *break* statement
- *switch* and *break* structure

Introduction

- **Nested If**
- **Logical operator**
 - And
 - Or
 - Not
- **Switch**

Complex Predicates (1/3)

- Logical expression and operators
 - AND (&&)
 - OR (||)
 - NOT (!)
 - Operand: 0 for false, **nonzero for true**
 - Predicate value: 0 for false, 1 for true
 - Truth table

Complex Predicates (2/3)

Operand_1	Operand_2	&&	
nonzero	nonzero	1	1
nonzero	0	0	1
0	nonzero	0	1
0	0	0	0

Complex Predicates (3/3)

- Precedence of logical operators

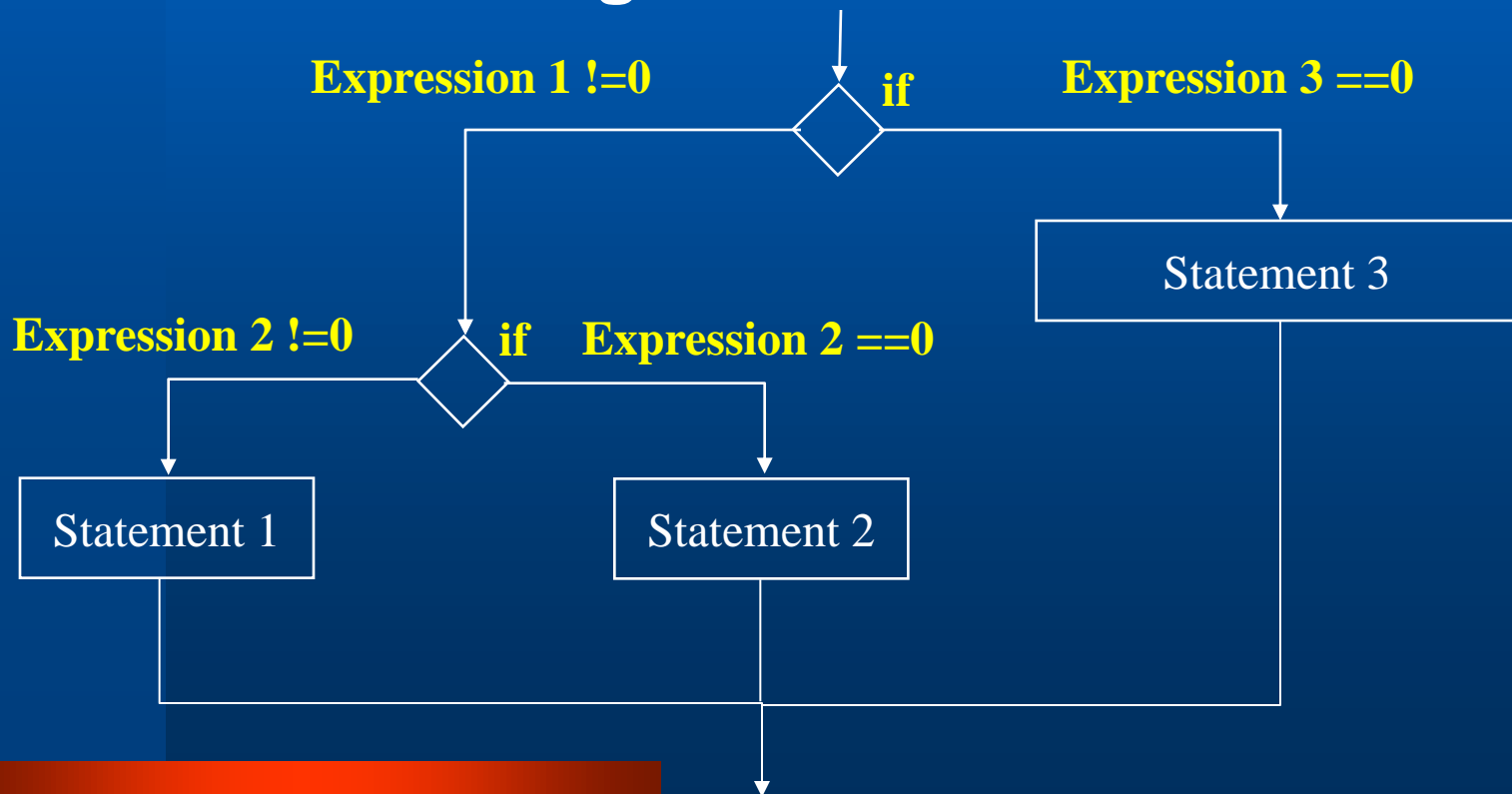
X=20, Y=4

If (X + y >= 13 && ! (x-y) || x * y - 16 == 4)

Step	Operator	Reduced expression
1	-	X + y >= 13 && ! 16 x * y - 16 == 4
2	!	X + y >= 13 && 0 x * y - 16 == 4
3	*	X + y >= 13 && 0 80 - 16 == 4
4	+	24 >= 13 && 0 80 - 16 == 4
5	-	24 >= 13 && 0 64 == 4
6	>=	1 && 0 64 == 4
7	==	1 && 0 0
8	&&	0 0
9		0

Complex Selection Structures (1/2)

- **Two-way selection**
 - Swim lane diagram



Complex Selection Structures (2/2)

```
if (Expression)
{
    if (Expression)
    Statement 1;
    else if
    Statement 2;
}
else if (Expression)
{
    Statement 3;
}
```


Example 1

- **General C**
- **Enter 1 to draw a rectangle, 2 to draw a triangle.**
 - Draw different rectangles.
- **06_c01.c**

Example 2

- **Robot Go!**
- **Robotic C: NXC**
- **If table is white, go forward; if table is black, rotate clockwise**
- **06_NXC01.nxc**

switch and *break* statement (1/2)

- Multi-way selection structure based on an integral value
- More clear than complex nested if
 - Three to ten alternatives

It becomes complicated when ten alternatives are available.

```
if (X + y >= 13 && ! 16 || x * y - 16 == 4){  
    Letter=A;  
    F=3*3+5  
    D=e+e*3}  
else if (X * y >= 13 && ! 16 || x * y + 16 == 4){  
    Letter=B;  
    F=3*4+5  
    D=e+e*5+3}
```

switch and *break* statement (2/2)

- Elements

- *switch*
- (expression)
- case clause
- *break* (optional)
- *default* (optional)

switch and *break* structure

```
void main ()  
{  
    switch (ControlExpression)  
    {  
        case number_1:  
        statements;  
        case number_2:  
        statements;  
        case number_3:  
        statements;  
    }  
}
```

Example 3

- **General C**
- **Enter 1 to draw a rectangle, 2 to draw a triangle.**
 - Draw different rectangles.
- **06_c02.c**

Example 4

- **Robot Go!**
- **Robotic C: NXC**
- **If table is white, go forward; if table is black and touch something, rotate clockwise**
- **06_NXC02.nxc**