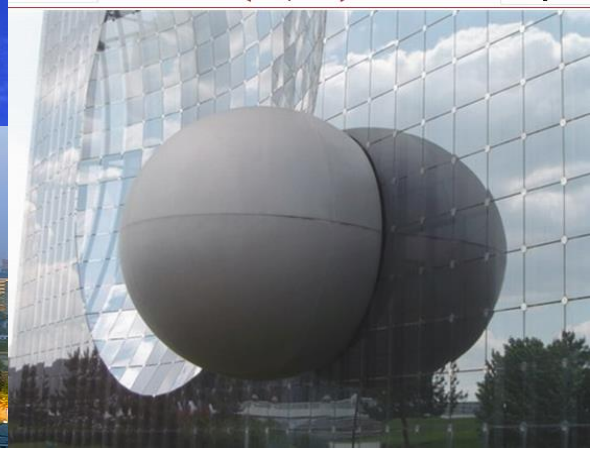


Computer Programming

Sino-European Institute of Aviation Engineering

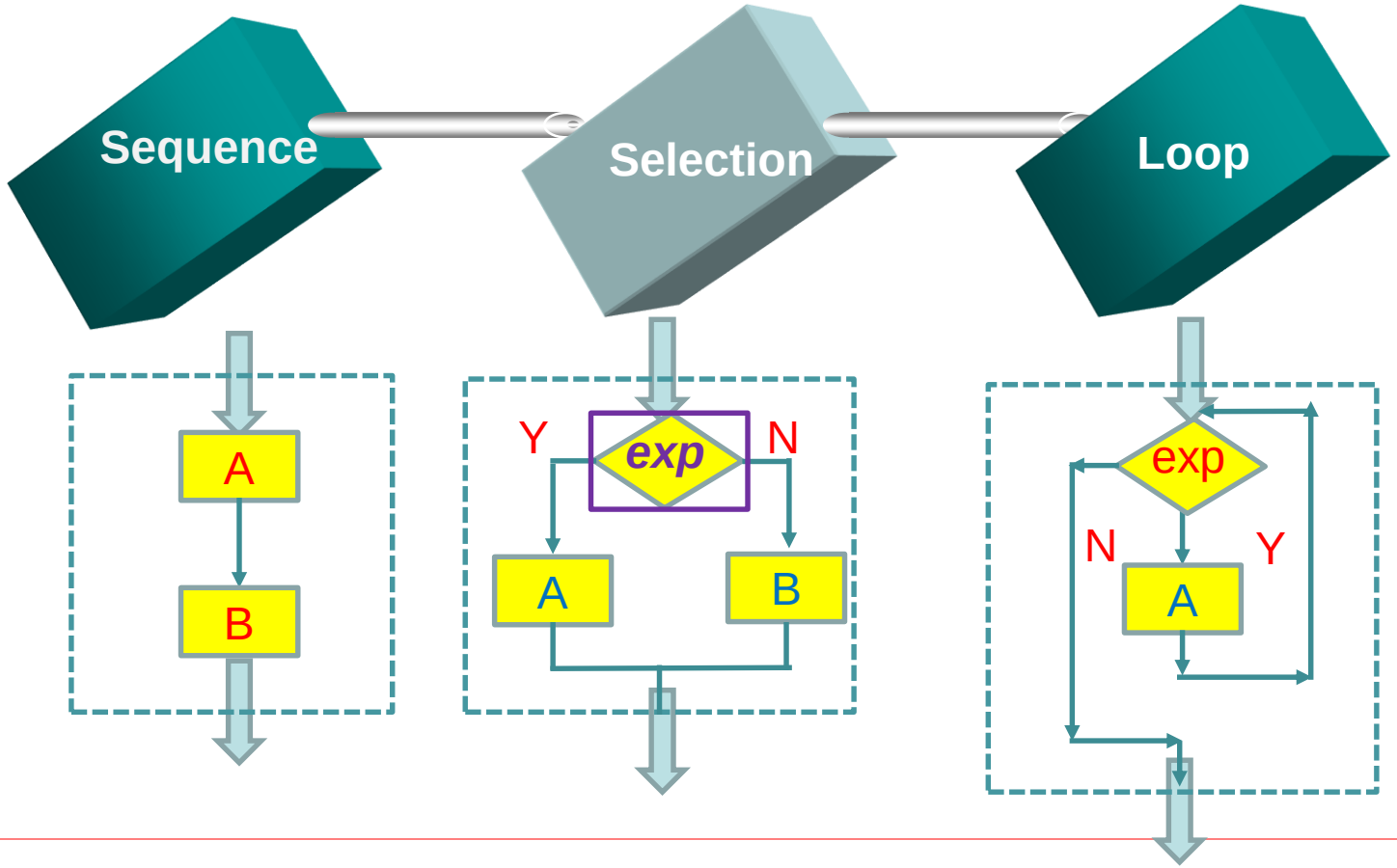
Module 3-2
Control Flow-Selection

Outline

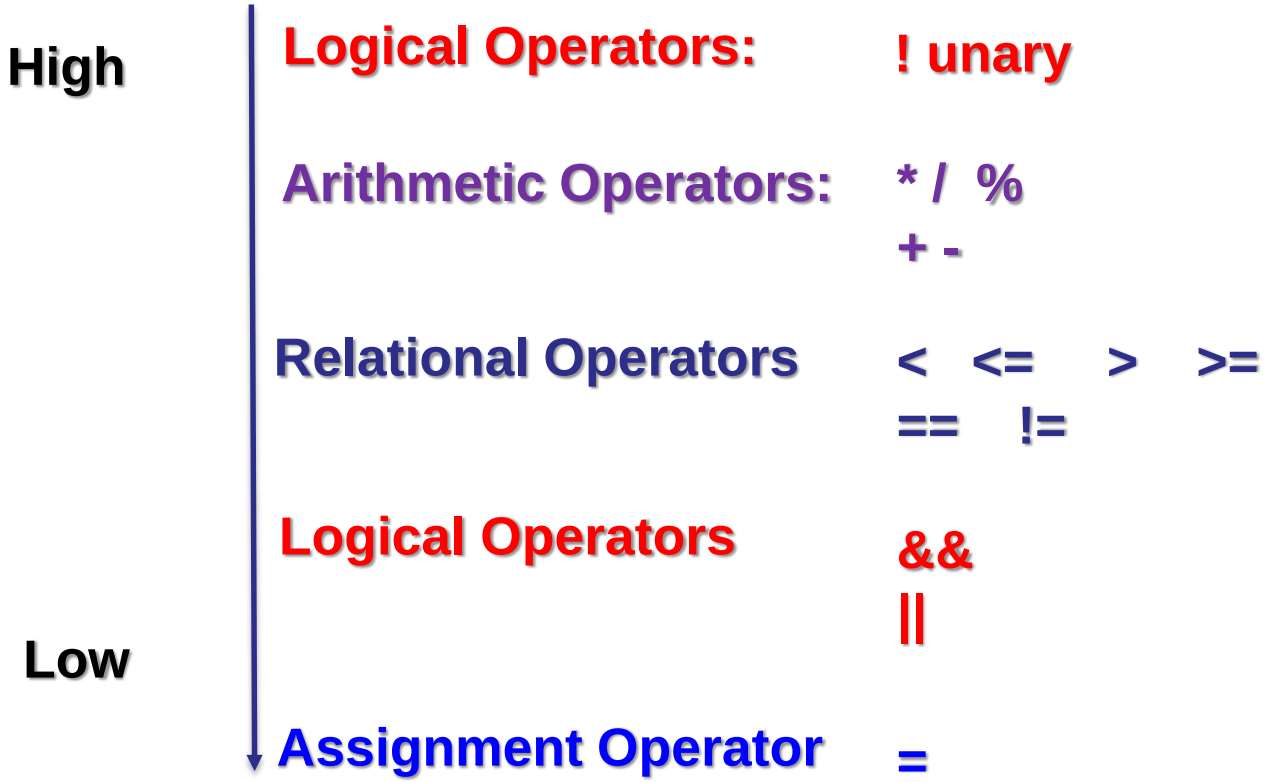
- Basic Control Flow
- Operators and Expressions
- The If Statement
- The Switch Statement

Basic Control Flow

The control-flow of a language specify the order in which computations are performed.



Operators and Expressions



The If Statement

Format

If (expression)

statement;

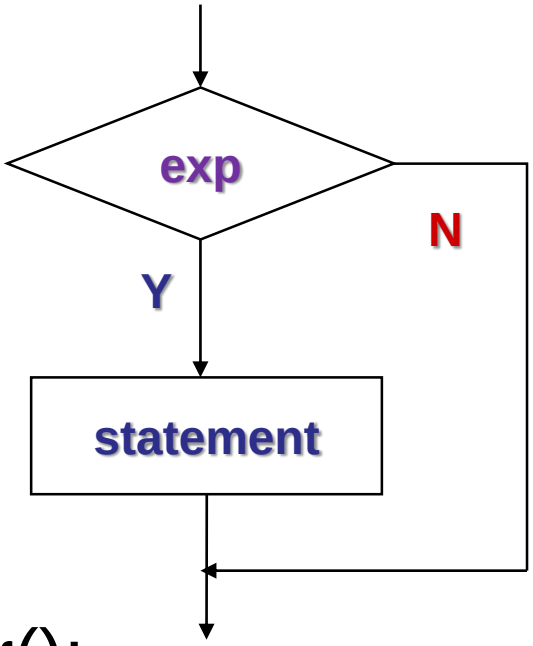
- Used to choose among alternative courses of action, only performs an action if the condition is true
- The expression can be any valid expression including a relational expression.

if (expression != 0) usually instead of *if (expression)*

The If Statement

- Three kinds of if statement

1. if（expression）
   statement$_1$



Ex:

```
ch=getchar();
if(ch>='a'&&ch<='z')
   ch=ch-'a'+'A';
  putchar();
```

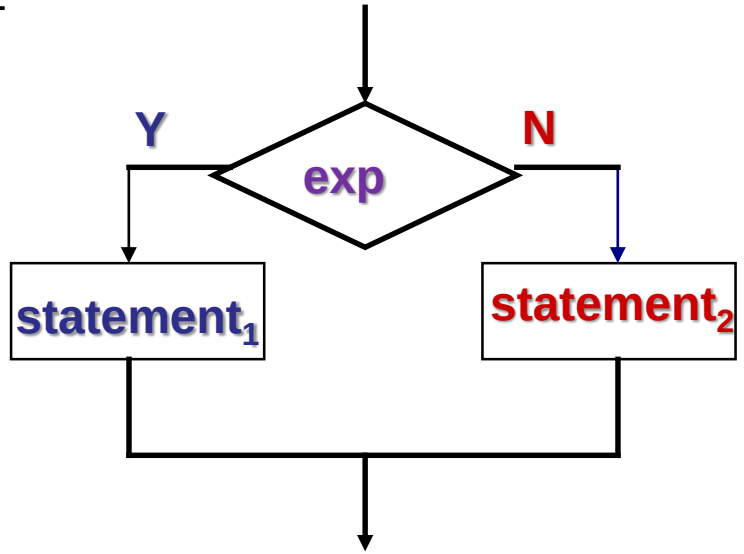
```
ch=getchar();
if(ch>='a'&&ch<='z')
  {ch=ch-'a'+'A';
   putchar();}
```

The If Statement

- Three kinds of if statement

2. if (expression) $statement_1$
   else $statement_2$

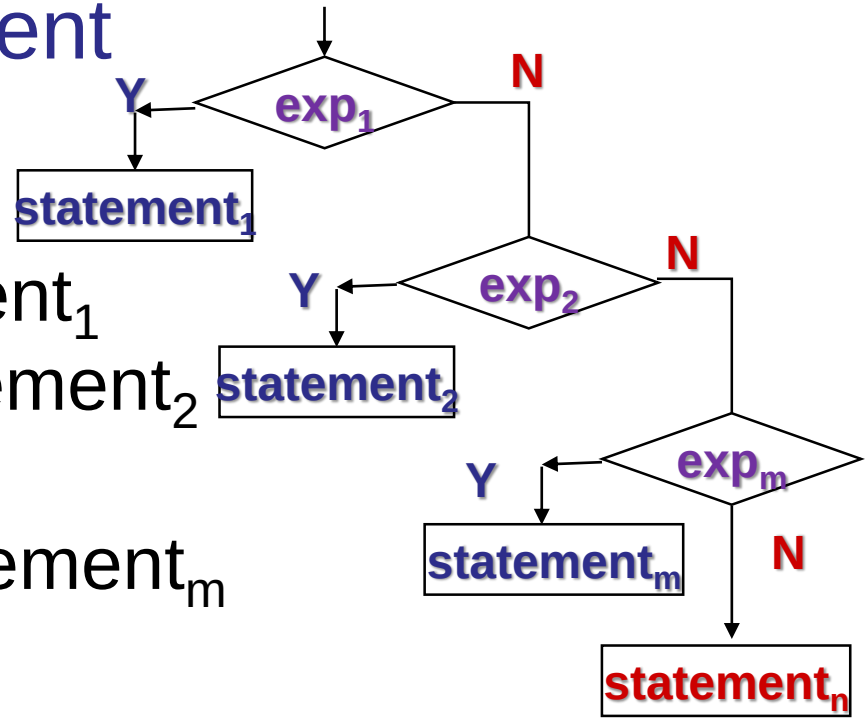
Ex:
```
if (x>y)
    printf("%d",x);
else
    printf("%d",y);
```



The If Statement

- Three kinds of if statement

3.if（expression$_1$）statement$_1$
else if (expression$_2$) statement$_2$
……
else if (expression$_m$) statement$_m$
else statement$_n$



The If Statement

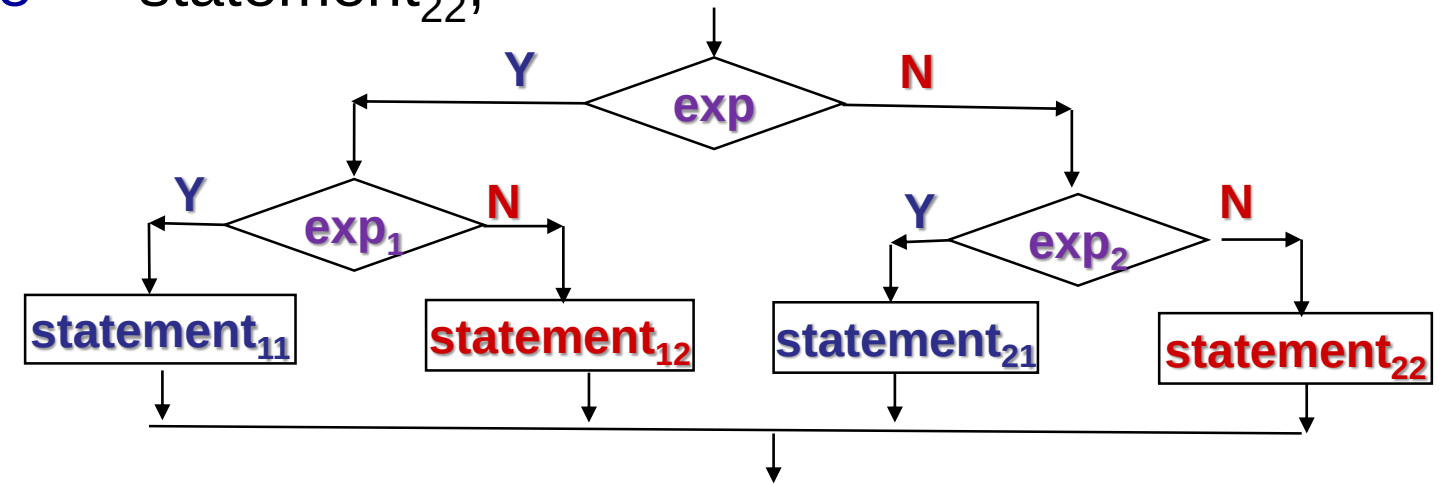
Notes

- This sequence of if statement is the most general way of writing a multi-way decision.
- The code for each statement is either single statement, or a group of them in braces.
- The last else part handles the “none of the above” or a default case where none of the other conditions is satisfied.

The If Statement

Nesting - If statement

```
if (exp)
    if (exp1)      statement11;
    else           statement12;
else if (exp2)     statement21;
else               statement22;
```



The If Statement

If(exp_1)
  if(exp_2) statement$_1$
  else statement$_2$

If(exp_1)
{ if(exp_2)
  statement$_1$}
else statement$_2$

Inner if

If(exp_1)
  statement$_1$
else
  if(exp_2) statement$_2$
  else statement$_3$

If(exp_1)
  if(exp_2) statement$_1$
  else statement$_2$
else
  if(exp_3) statement$_3$
  else statement$_4$

The If Statement

Right or Wrong?

$$y=\begin{cases} -1 & (x<0) \\ 0 & (x=0) \\ 1 & (x>0) \end{cases}$$

1：

```
if(x<0)
    y=-1;
else
    if (x==0)  y=0;
    else  y=1;
```

✓

2：

```
if (x>=0)
    if(x>0)  y=1;
    else     y=0;
else   y=-1;
```

✓

3:

```
y=-1;
if (x!=0)
    If(x>0)  y=1;
else  y=0;
```

✗

The If Statement

```
#include <stdio.h>
void main()
{
   int digit, i, letter, other;   char ch;
   digit = letter = other = 0;
   printf("Enter 10 characters: ");
   for(i = 1; i <= 10; i++)
    { ch = getchar();     /* Input a character*/
      if((ch >= 'a' && ch <= 'z' ) || ( ch >= 'A' && ch <= 'Z'))
           letter ++;
      else if (ch >= '0' && ch <= '9')
           digit ++;
      else
           other ++;
     }
   printf("letter=%d,digit=%d,other=%d\n",letter,digit,other);
}
```

input 10 characters: Hello123?

letter=5, digit=3, other=2

The Switch Statement

Switch

- a multi-way conditional statement
- Useful when a variable or expression is tested for all the values it can assume and different actions are taken
  - similar to if-else
  - allows the selection of an arbitrary number of choices based on an integer value

The Switch Statement

Format

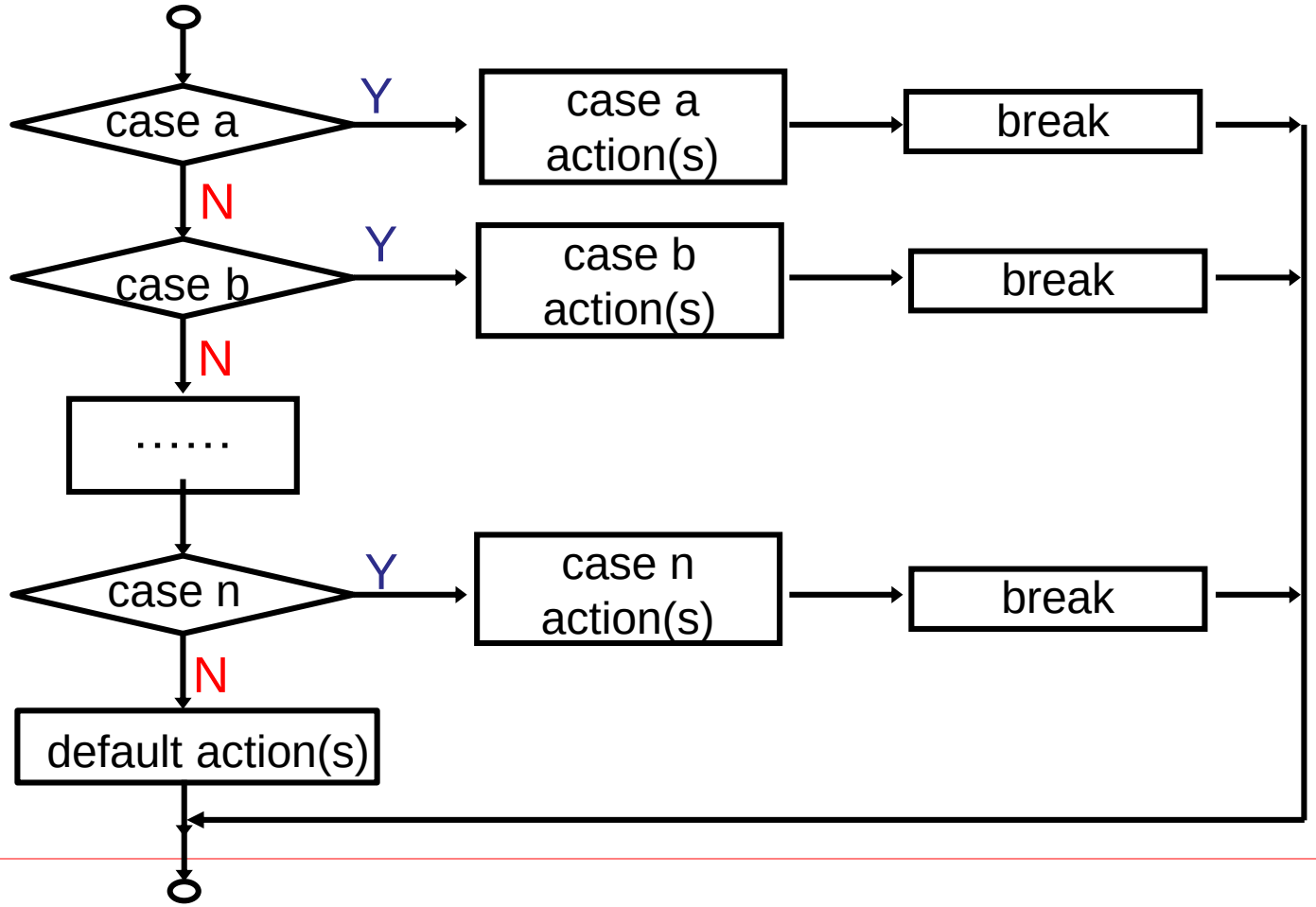
- Series of case labels and an optional default case

```
switch (expression)
{
    case  constant1:
            actions; break;
    case constant2:
            actions; break;
    default:
            actions;
}
```

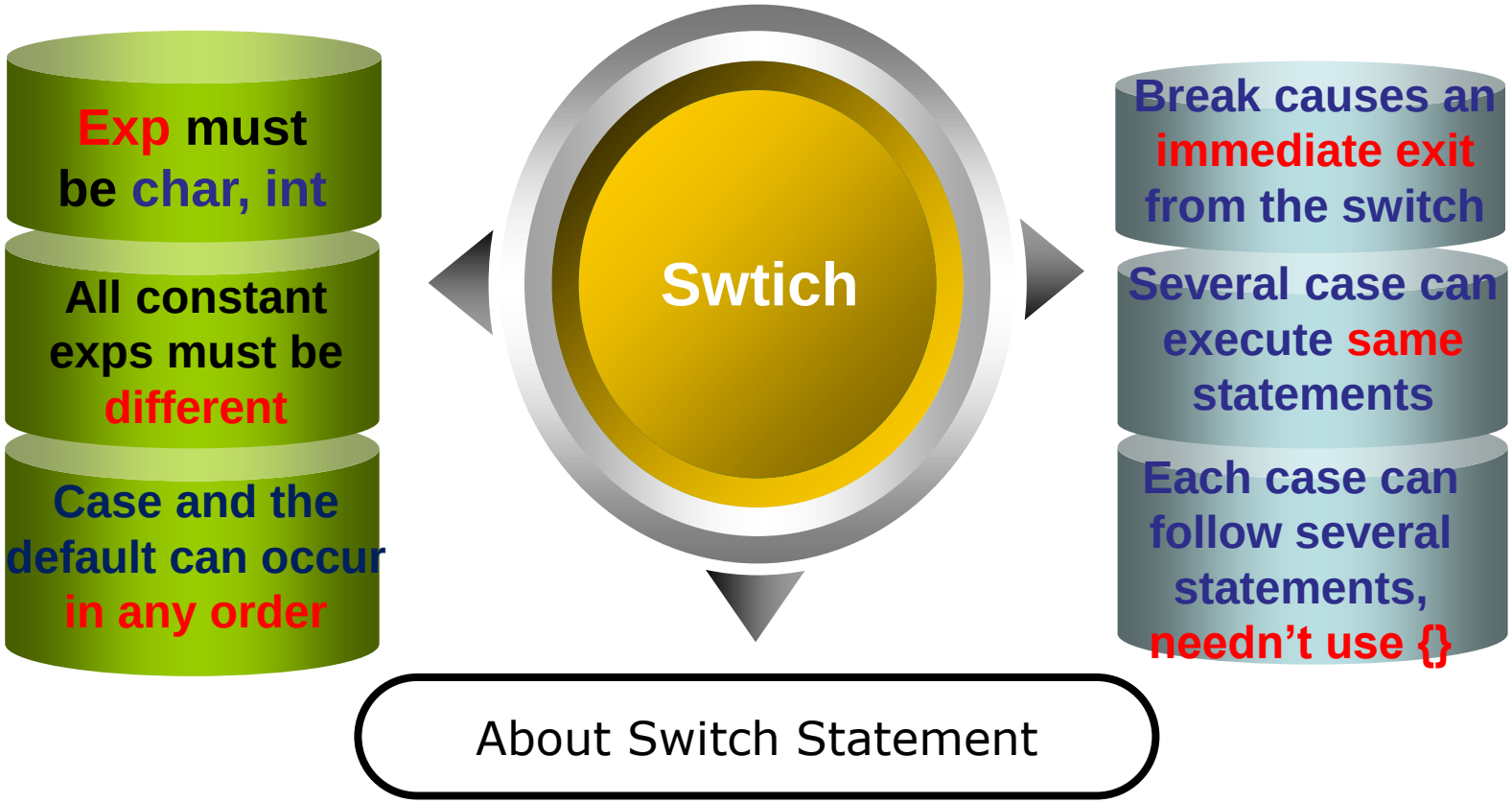
- break: exits from structure

The Switch Statement

- Flow chart of switch structure



The Switch Statement



The Switch Statement

Notes

- when the switch transfers to the chosen case, it starts executing statements at that point
- it will "fall through" to the next case unless you "break out"
- break causes the program to immediately jump to the next statement after the switch statement

The Switch Statement

- Given your ID, return the test score

```
int id, score;
...
switch (id)
 {
   case 1:
      score = 90;
   case 2:
      score = 100;
   case 3:
      score = 87;
   default:
      score = 0;
}
printf("Your test score is %d\n", score);
```

The Switch Statement

- Without break

```
        if (id == 1) jump to a;
        if (id == 2) jump to b;
        if (id == 3) jump to c;
        jump to d;

a:      score = 90;

b:      score = 100;

c:      score = 87;

d:      score = 0;

end:
```

Code executes sequentially

The Switch Statement

- With break

```
        if (id == 1) jump to a;
        if (id == 2) jump to b;
        if (id == 3) jump to c;
        jump to d;

a:      score = 90;
        jump to end;
b:      score = 100;
        jump to end;
c:      score = 87;
        jump to end;
d:      score = 0;

end:
```

The Switch Statement

```
# include <stdio.h>
void main()
{
    char operator;   double value1, value2;
    printf("Type in an expression: ");
    scanf("%lf%c%lf", &value1, &operator, &value2);
    switch(operator)
    {
        case '+':
            printf("=%.2f\n", value1+value2); break;
        case '-':
            printf("=%.2f\n", value1-value2); break;
        case '*':
            printf("=%.2f\n", value1*value2); break;
        case '/':
            printf("=%.2f\n", value1/value2); break;
        default:
            printf("Unknown operator\n"); break;
    }
}
```

```
Type in an expression: 3.1+4.8
=7.90
```

The Switch Statement

Exercise

Write a program that accepts a number between 1 and 100 from the user. If there is a coin of that value in cents, it should display its name. Otherwise, it should report that there is no such coin

1 = cent, 5 = nickel, 10 = dime, 25 = quarter, 100 = dollar
Remember to check for the validity of the input!

The Switch Statement

```
# include <stdio.h>
int main()
{
    int num;
    printf("Please enter a number from 1 to 100: ");
    scanf("%d", &num);
    if (num < 1 || num>100)
    {
       /* Make sure the input is valid*/
        printf("Invalid input!\n");
        return 1;
    }
    switch (num)
    {
        /* Display the correct coin name, or a default message if there's no such coin*/
        case 1:
            printf("It's a cent!\n");
            break;
```

The Switch Statement

```
        case 5:
            printf("It's a nickel!\n");
            break;
        case 10:
            printf("It's a dime!\n");
            break;
        case 25:
            printf("It's a quarter!\n");
            break;
        case 100:
            printf("It's a whole dollar!\n");
            break;
        default:
            printf("It's not a coin!\n");
    }
     return 0;
}
```

Summary

- Basic control flow
  - Sequence
  - Selection
  - Loop
- The if statement
  - Three kinds of the If statement
    - if
    - If~else
    - If~ else if ~ else
  - Nestting
- Switch
  - Case
  - Break

Thank you!