

CSE 142 Computer Programming I

Switch Statement

© 2000 UW CSE

J-1

Overview

Concepts this lecture

The switch statement

Choosing between if and switch

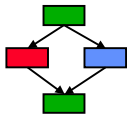
Reading

Textbook sec. 4.8

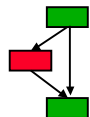
J-2

Review: Conditional Control Flow

The if statement chooses one of two statements to execute before continuing



An if statement could also be used to decide whether or not to skip a statement before continuing

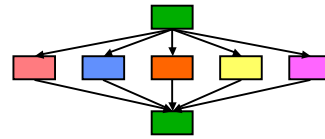


J-3

Multi-way Control Flow

The choice may be “multi-way” rather than simply between two alternatives

In C, if statements can be used, and sometimes a statement called the switch can be used



J-4

Multi-way Choice with *if*

/ How many days in a month? */*

```
if ( month == 1 ) {           /* Jan */
    days = 31 ;
} else if ( month == 2 ) {    /* Feb */
    days = 28 ;
} else if ( month == 3 ) {    /* Mar */
    days = 31 ;
} else if ( month == 4 )     /* Apr */
    days = 30 ;
...                          /* need 12 of these */
```

Better...

```
if ( month == 9 || month == 4 || /* Sep, Apr */
    month == 6 || month == 11 ) { /* Jun, Nov */
    days = 30 ;
} else if ( month == 2 ) {       /* Feb */
    days = 28 ;
} else {
    days = 31;                   /* All the rest */
}
```

J-6

Alternative: switch

A switch is a form of conditional statement.

It is specifically designed to be useful in multi-way choice situations.

Instead of a condition, there is a value which is tested, and a series of **cases** of which only one may be chosen.

J-7

Using switch

```
/* How many days in a month? */
switch ( month ) {
case 2:          /* February */
    days = 28 ;
    break ;
case 9:          /* September */
case 4:          /* April */
case 6:          /* June */
case 11:         /* November */
    days = 30 ;
    break ;
default:        /* All the rest have 31 ...*/
    days = 31 ;
}
printf ( "There are %d days. \n ", days ) ;
```

J-8

Symbolic constants work, too

```
/* How many days in a month? */
#define JANUARY 1
#define FEBRUARY 2
...
#define DECEMBER 12

switch ( month ) {
case FEBRUARY:
    days = 28 ;
    break ;
case SEPTEMBER:
case APRIL:
case JUNE:
case NOVEMBER:
    days = 30 ;
    break ;
default:          /* All the rest have 31 ...*/
    days = 31 ;
}
printf ( "There are %d days. \n ", days ) ;
```

J-9

switch Statement

The syntax of switch differs from other C statements

```
switch (int expression) {
...
/*a series of cases */
...
}
```

The value of the expression determines which of the cases is executed.

J-10

Cases

A case is a section of code within the switch statement. A case is executed only if the switch expression has a specified value

case value:
/ a sequence of statements*/*

The sequence is typically ended with special statement

break;

break causes the entire switch statement to end

J-11

The switch Expression

The switch expression is *not* a conditional expression as it is in an *if* statement

Only an integer expression is allowed

Most often, the expression is a single integer variable

The value of the variable determines which case is chosen

J-12

switch: Flow of Control

```
month = 6 ;
switch ( month ) {
case 2:          /* February */
    days = 28 ;
    break ;
case 9:          /* September */
case 4:          /* April */
case 6:          /* June */
case 11:         /* November */
    days = 30 ;
    break ;
default:        /* All the rest have 31 ... */
    days = 31 ;
}
printf ( "There are %d days. \n ", days ) ;
```

The Biggest Pitfall of switch

```
month = 6 ;
switch ( month ) {
case 2:          /* February */
    days = 28 ; /* break missing */
case 9:          /* September */
case 4:          /* April */
case 6:          /* June */
case 11:         /* November */
    days = 30 ; /* break missing */
default:        /* All the rest have 31 ... */
    days = 31 ;
}
printf ( "There are %d days. \n ", days ) ;
```

switch on char is also legal

```
char marital_status ;
...
switch ( marital_status ) {
case 'm':
case 'M':
    printf ( "Married \n" ) ;
    break ;
case 's':
case 'S':
    printf ( "Single \n" ) ;
    break ;
default:
    printf ( "Sorry, I don't recognize that code. \n" ) ;
}
```

int or char expression

Remember... a char is just an ASCII value underneath!

Summing Up

Switch is a form of conditional statement

Switch works for multi-way conditions that depend upon an integer (or char) value

Beware the syntax of switch!

The switch and if statements are *not* fully interchangeable

J-16

Bonus Footnote

```
char marital_status ;
...
switch ( marital_status ) {
case 'm':
case 'M':
...
}
```

Why should a character be allowed here, when the expression is supposed to be an integer?

Answer: The actual machine representation of a character is a small integer.

Most of the time, however, you should treat ints, and chars as fully different types!

QOTD: A switch Minefield

Explain everything that's wrong with the following ill-conceived switch statement...

```
switch (some_num) {
printf("This line will NEVER be executed! Why?\n");
case 2:
    printf("Something simple wrong with this!\n");
case BASE != 36:
    printf("This will work, but not how you expect!\n");
    break;
case 4:
    printf("Syntactically valid..\n");
    printf("But it won't do what you want it to do!\n");
    break;
default:
    printf("This one's syntactically valid, too!\n");
}
```

J-18