

Chapter 9

More Control Structures

Switch Statement

- A multiway branching control structure
- An alternative to nested if statements
- Uses a switch expression and cases to determine which branch to take
- Switch Expression:
 - The expression whose value determines which switch label is selected. It cannot be a floating-point or a string.

Example

```
switch ( letter )
{
    case 'X': Statement1;
              break;
    case 'L':
    case 'M': Statement2;
              break;
    case 'S': Statement3;
              break;
    default:  Statement4;
}
Statement5;
```

Details

- The switch statement must be of constant type, i.e., char, short, int, long, bool, or enum.
- What comes after the case label must match in type either directly or through coercion to the type of the switch expression.
- The default label is optional.

switch vs if

```
switch ( grade )
{
    case 'A':
    case 'B': cout << "Good Work";
              break;
    case 'C': cout << "Average Work";
              break;
    case 'D':
    case 'F': cout << "Poor Work";
              numberInTrouble++;
              break;
}
```

switch vs if

```
if ( grade == 'A' || grade == 'B' )
    cout << "Good Work";
else if ( grade == 'C' )
    cout << "Average Work";
else if ( grade == 'D' || grade == 'F' )
{
    cout << "Poor Work";
    numberInTrouble++;
}
```

Do-While Loop

- The do while loop is almost the same as the while loop
- The difference is the loop condition is tested at the bottom of the loop
- This behavior ensures that the do-while loop is always executed at least once.

Example

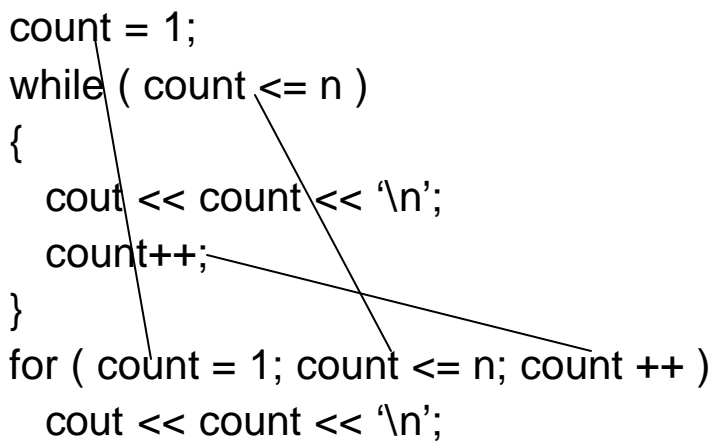
```
do
{
    cout << "Enter your age: ";
    cin >> age;
    if ( age <= 0 )
        cout << "Your age must be positive.\n";
} while ( age <= 0 );
```

For Loop

- The For statement is designed to simplify the writing of count controlled loops
- You simply take all the pieces you had in a while loop and stick them in the slots in the for loop

Example

```
count = 1;
while ( count <= n )
{
    cout << count << '\n';
    count++;
}
for ( count = 1; count <= n; count ++ )
    cout << count << '\n';
```



Break and Continue

- The break statement can be used with any loop and switch statements.
- It causes the inner most control structure to be exited.
- Break can be used to simplify logic, but should be used only as a last resort, not a daily occurrence.
- Continue can be used only with loops
- It causes the loop to skip to the bottom of the loop iteration and begin a new loop iteration.

Loop Choices

1. If the loop is a count-controlled loop, the For loop is a natural choice.
2. If the loop is an event-controlled loop whose body should execute at least once, choose a do-while loop.
3. If the loop is an event-controlled loop and nothing is known about the first execution, use a while loop
4. When in doubt, use a while loop.
 - Pages 453-454.