

Chapter 6

Loops

- A control structure that causes a statement or group of statements to be executed repeatedly
- There are 3 types of loops
 - while
 - for
 - do while

The While Statement

```
while (expression)  
    Statement
```

```
while (inputVal != 25)  
    cin >> inputVal;
```

while and if

- Not the same construct
- If statements do not repeat
- While statements will repeat until the expression becomes false
- While statements should always have a statement that will eventually make the expression to become false

Phases of Loop Execution

- Loop entry – The point at which the flow of control reaches the first statement inside a loop.
- Iteration – An individual pass through or repetition of, the body of a loop.
- Loop test – The point at which the While expression is evaluated and a decision is made either to begin a new iteration or skip to the statement immediately following the loop.
- Loop exit – the point at which the repetition of the loop ends and control passes the first statement following the loop.
- Termination condition – The condition that causes the loop to be exited.

Loops using the While Statement

- Count-Controlled Loops
- Event-Controlled Loops
 - Sentinel-Controlled Loops
 - End-of-File Controlled Loops
 - Flag-Controlled Loops

Count Controlled Loops

```
loopCount = 1;
while (loopCount <= 10)
{
    cout << "Hello!\n";
    loopCount++;
}
```

Sentinel-Controlled Loops

```
while(!(month == 2&&day == 31))
{
    cin >> month >> day;
    ...
}
```

Sentinel-Controlled Loops

```
cin >> month >> day;
while(!(month == 2&&day == 31))
{
    cin >> month >> day;
    ...
}
```

Sentinel-Controlled Loops

```
cin >> month >> day;
while(!(month == 2&&day == 31))
{
    ...
    cin >> month >> day;
}
```

End-of-File Controlled Loop

```
cin >> month >> day;
while ( cin )
{
    ...
    cin >> month >> day;
}
```

End-of-File Controlled Loop

```
cin >> month >> day;
while ( !cin.eof() )
{
    ...
    cin >> month >> day;
}
```

Flag-Controlled Loop

```
sum = 0;
cin >> number;
bool nonNegative = true;
while ( nonNegative )
{
    if ( number > 0 )
        sum = sum + number;
    else
        nonNegative = false;
    cin >> nonNegative;
}
```

Quiz

- Write a code fragment that uses a loop to compute the sum of the squares of a series of numbers from 1 to a user entered number. Upon loop exit the sum of the squares should be printed.


Looping Subtasks

- Counting
- Summing
- Keeping track of the previous value

Counting Loop

```
count = 0;
cin.get( inChar );
while ( inChar != '.' )
{
    count++;
    cin.get( inChar );
}
```

Iteration Counter: A counter variable that is incremented with each iteration of a loop



Summing Loop

```
sum = 0;
cin >> number;
bool nonNegative = true;
while ( nonNegative )
{
    if ( number > 0)
        sum = sum + number;
    else
        nonNegative = false;
    cin >> nonNegative;
}
```

Remembering Loop

```
count = 0;
in.get ( prevChar );
in.get ( currChar );
while ( !in.eof() )
{
    if ( currChar == '=' && prevChar == '!' )
        count++;
    prevChar = currChar;
    in.get( currChar );
}
```

Designing Loops

- What is the condition that ends the loop?
- How should the condition be initialized?
- How should the condition be updated?
- What is the process being repeated?
- How should the process be initialized?
- How should the process be updated?
- What is the state of the program on exiting the loop?