For loop

Another looping techniques

While/Do ... While

- Good looping techniques but limited
  - Have to cater for counter
  - Terminating condition can be difficult to implement
- For loop has in-built
  - Start position
  - Terminating condition
  - Increment/Decrement

For loop - Components

```
for ( start_value; end_value;
   increment_number )

{

//YOUR_CODE_HERE

} // End of for loop
```

- The word for
- ( )'s
- ; seperating the 3 arguments
- { }'s

For loop example

A simple for loop that will print 0 to 9

```
for(int i = 0;i<10;i++)
{
System.out.println(i);
}
```

Steps

Taking the last example ....

```
1. for(int i = 0;i<10;i++)
2. {
3. System.out.println(i);
4. }
5. // Beyond the for loop
```

1. Start the counter i at 0 – FIRST PARAMETER

2. Check if the test is false – SECOND PARAMETER

3. Do whatever is inside the { }'s

4. Then increment (++) or decrement (--)

5. Go around and check if the test is still true

   1. If TRUE, do the next iteration

   2. If FALSE, then go past the for loop (Line 5)

Typical Uses of for loop

- A means to count up/down at a specific interval

- Loading or traversing through

  - Strings

  - Arrays

  - ArrayLists

For loop – What to be aware of ...

- If no { }'s
  - It will compile
  - BUT will only control next line or block
  - Rare but technically feasible
- Scope
  - If variable declared within for loop, it can only be visible/used within that loop OR any construct used within that
- Placing of { }'s
  - Especially if placing a loop into existing code

Break and Continue

- Special functions to be used if needing to
  - Stop the loop entirely
    - break;
  - Go to the next iteration of the loop
    - continue;
- Break is understandable to use, but why Continue?
  - Rarely within one loop
  - But in nested loops with labels ....

For ... each loop

- Some texts will refer to a new loop

  - For ... each loop

- Used primarily for processing elements within an **array**

Links

- http://www.homeandlearn.co.uk/java/java_for_loops.html