Control Flow 2

Control flow
while & do-while
for
break, continue, return
The **while** & **do-while** statements

- The **while** statement continuously executes a block while a condition is **true**.
- The **do-while** construct evaluates the condition at the end of the block rather than at the start.

Imperative programming: sequence, selection, **iteration**.
The **for** statement

- A compact way to *iterate* over a set of values.
- The statement has three logical parts:
  - Initialization
  - Termination condition
  - Increment statement
- The ‘enhanced’ **for** statement *infers* the initialization, termination and increment statements, given an array or collection
Branching statements

- The **break** statement terminates a loop construct
  - *Unlabeled* terminates the loop in which it is called
  - *Labeled* terminates the loop named by the label
- The **continue** statement skips the current iteration of a loop
  - *Unlabeled* skips the current iteration of the loop in which it is called
  - *Labeled* skips the current iteration of the loop named by the label
- The **return** statement exits the current method