10 Control Flow - Repetition

while (condition) {
  statements
}

for (init; condition; update) {
  statements
}

while (condition) {
  do {
    statements
  }
  while (condition);
}

for variable in sequence:

  statements

Anatomy of Loop

1. Loop index/loop control variable
2. Loop condition
3. Loop update
4. Loop body

- break
- continue

Other Loop Designs

while (1) {
  if (condition)
  break;
}

Indefinite Loops

while True:
  if condition:
    break
for (; ; ; ) {
  if (condition)
    break
}
while True:
    for ; ; ; {
        ...                             ...
        ...                        ...
        if (condition):
            break;
Loop Design Issues

- Loop control variable (also optional in for-loop).
- Scope of the LCV

```python
for i in range(10):
    print(i)
    i = i + 1
```

- Can the LCV be modified in the loop?

- Other Languages

```
for i := 0 to n-1 do
begin
    statements
    count_up
end

for i := n-1 downto 0 do
begin
    statements
    count_down
end
```

```
FOR i := 0 TO n-1 DO
    statements
END

FOR i := n-1 TO 0 BY -1 DO
    statements
END
```
What happens at edge cases?

```java
for (int i = -1; i <= 2147483647; i++)
```

2147483646
+1
--------------
2147483647
+1
--------------
2147483648

C/C++: INT_MIN, INT_MAX
Java: Integer.MAX_VALUE, Integer.MIN_VALUE

- Does the PL allow iteration over user-defined types?

```java
BinaryTree<T> t;
for (T item : t)
```

```java
public class BinaryTree<T>
    implements Iterator<T>
```

```java
int[] a = ...;
for (int item : a)
```

```
// can use the value of a
```
```
// but not modify it.
```
names = ["Bryn Mawr", "Havenford", "Swarthmore", ...]

for name in names:
    word = "Bryn Mawr"
    for w in word:
        print(w)

Python Iterators are co-routines.

Java: User-defined objects/classes have to implement their own iterators:

1. Implement Iterable interface
   ```java
   public Iterator iterator();
   ```

2. Interator interface
   ```java
   public boolean hasNext();
   public E next();
   public void remove();
   ```

ArrayList<Place> places... 

```java
Iterator<Place> iter = places.iterator();
while (iter.hasNext()) {
    Place p = iter.next();
}
```

for (Place p : places)
### 10 Control Flow - Recursion

**Monday, October 19, 2020  8:25 AM**

**Iterative Code**

\[ Sum = \sum_{i=1}^{n} i \]

```c
int sum(int low, int high) {
    int result = 0;
    for (int i=low; i <= high; i++)
        result = result + i;
    return result;
} // sum()
```

```c
int gcd(int a, int b) {
    while (a != b) {
        if (a > b)
            a = a - b;
        else
            b = b - 1;
    }
    return a;
} // gcd()
```

```c
int factorial(int n) {
    int result = 1;
    for (int i=1; i <= n; i++)
        result *= i;
    return result;
} // factorial()
```

**Recursive Code**

```c
int sum(int low, int high) {
    if (low == high)
        return low;
    else
        return low + sum(low+1, high);
} // sum()
```

```c
int gcd(int a, int b) {
    if (a == b)
        return a;
    else if (a > b)
        return gcd(a-b, b);
    else
        return gcd(a, b-a);
} // gcd()
```

```c
int factorial(int n) {
    if (n == 0)
        return 1;
    else
        return n * factorial(n-1);
} // factorial()
```

**Tail Recursion**

**Tail Call Optimization (TCO)**
factorial(n)

n = 3

factorial(n)

n = 4

4 * factorial(3)

main

factorial(n)