

C++ Loops: do/while

• Similar to while loop

- Loop condition
- Loop body

Note:  
do/while

• But also different! loops are less common

- Condition test at end
- Body executes at least once

---

---

---

---

---

---

---

---

Loop Example: do/while

```
int x = 6;
const int firstInt = 1;
int upperLimit = x;
int oddSum = 0;
int nextOdd = firstInt;

do
{
  oddSum += nextOdd;
  nextOdd += 2;
} while (nextOdd <= upperLimit)
```

---

---

---

---

---

---

---

---

Loop Structure: while

```
initialize          A reminder
startup
while (more)
{
  process current item
  next
}
finalize
```

---

---

---

---

---

---

---

---

C++ Loops: for

- Loop construction shorthand
- Equivalent to while loop
  - Condition ("more"), body ("process")
- But also incorporates:
  - Setup ("initialize/startup")
  - Iteration ("next")

Most popular C++ loop

---

---

---

---

---

---

---

---

Loop Operation: for

Equivalent loops

```
<init>;
while (<more>)
{
  <process>;
  <next>;
}
```

```
for (<init>;
     <more>;
     <next>)
{
  <process>;
}
```

---

---

---

---

---

---

---

---

Loop Example: for

```

      <init>   <more>   <next>
int num; ↓     ↓       ↓
for ( (num=1; num <= 5; num += 1)
{
  cout << num << endl;
}
      ↑
    <process>

```

1  
2  
3  
4  
5

---

---

---

---

---

---

---

---

### Increment/Decrement Operators

- **Already know side effects**
  - Increment: add one (++)
  - Decrement: subtract one (--)
- **Also return a value**
  - Prefix: value after modification
  - Postfix: value before modification

---

---

---

---

---

---

---

---

### Increment Example

```

int a = 3;
int b = 7;
int x;
int y;

x = a++;
y = ++b;

```

a:   
b:   
x:   
y:

---

---

---

---

---

---

---

---

### Loop With Increment Operator

```

      <init>   <more>   <next>
int num; ↓     ↓       ↓
for ( (num=1; num <= 5; num++) )
{
  cout << num << endl;
}
      ↑
    <process>

```

1  
2  
3  
4  
5

---

---

---

---

---

---

---

---

Comma Operator

- **A strange operator**
  - Might be better if not invented
- **Not all commas are operators**
  - E.g., argument separators
- **But some commas are**
  - Separate two expressions

---

---

---

---

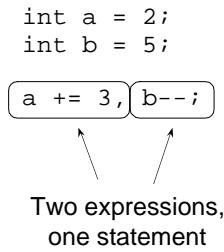
---

---

---

---

Comma Operator Example




---

---

---

---

---

---

---

---

Comma Operators in for Loop

```
int exp = 5; // max exponent
int n; // exponent
int p; // result
for (n=0, p=1; // init
     n <= exp; // more
     n++, p *= 2) // next
{
  cout << n << " " << p << endl;
}
Print table of powers of two
```

---

---

---

---

---

---

---

---

break Statement

- **Have seen it with switch**
  - Exits from switch body
  - Terminates current case
- **Also permitted in loops**
  - Causes immediate exit from loop
  - Might be better off without it
    - Put exit condition in loop condition

---

---

---

---

---

---

---

---

Loop break Example

```
double num;
for (int k=0; k<10; k++)
{
  cin >> num;
  if (num < 0.0) break;
  cout << sqrt(num) << endl;
}
cout << "The End" << endl;
```

Loop ten times, but stop on negative value

---

---

---

---

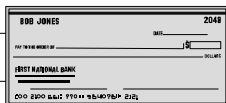
---

---

---

---

Design Exercise



- **Loan amortization table**
- **Given:**
  - Original principal amount
  - Annual interest rate
  - Monthly payment
- **Print table of monthly values**
  - Month, interest, old/new balance

---

---

---

---

---

---

---

---

⋮  
**Design Exercise Output**

Annual interest = 12%  
Monthly payment = 200.00  
Month Balance Interest New Bal  
1 5000.00 50.00 4850.00  
2 4850.00 48.50 4698.50  
3 4698.50 46.99 4545.49  
...

Stop when balance becomes negative  
(loan paid off).

---

---

---

---

---

---

---

---