Slides #9

for Loop

CMPT 130

© Dr. B. Fraser

for

- Definite Loop:
  - A loop where we know ...

    “Count from 1 to 10..”
- Indefinite Loop:
  - A loop where we...
    tell how many times we will execute the loop:
    “Count up from 1 to find first multiple of 3, 4 and 18”
- for loops are often useful to neatly organizing definite loops.

Simple Example

```
for (int i = 0; i < 5; i++) {
    // Do amazing stuff here!
    cout << "i: " << i << endl;
}
```

Output:

```
i: 0
i: 1
i: 2
i: 3
i: 4
```

Above code identical to:

```
{
    int i = 0;
    while (i < 5) {
        cout << "i: " << i << endl;
        i++;
    }
}
```

Which is easier to read?

```
for (int i = 0; i < 5; i++) {
    cout << "i: " << i << endl;
}
```

```
{
    int i = 0;
    while (i < 5) {
        cout << "i: " << i << endl;
        i++;
    }
}
```

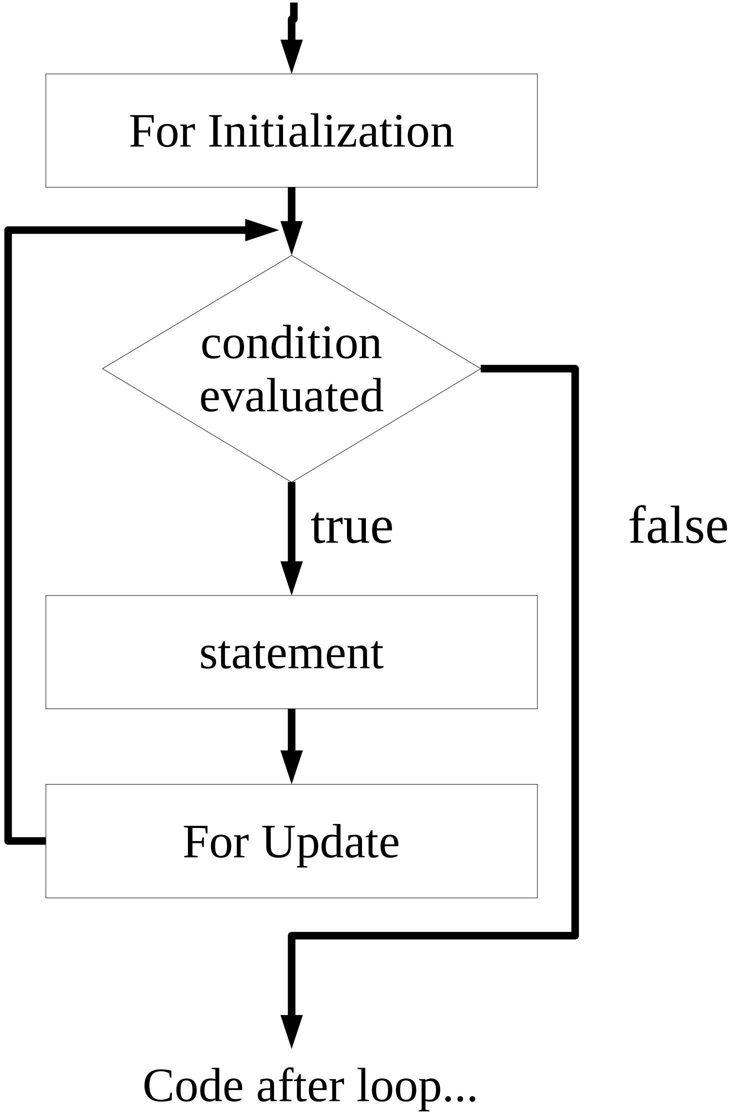
Work Through Examples

```
for (int i = 0; i < 12; i++) {
        cout << "Month # " << i << endl;
}


for (int i = 100; i > 0; i--) {
        cout << i << endl
}


for (int rows = 0; rows < 10; rows++) {
        if (rows % 3 == 0) {
                rows ++;
        }
        cout << rows;                    ..
}
```

```
for (int rows = 0; rows < 10; rows++) {
    if (rows % 3 != 0) {
        cout << rows;
    }
}
```



Notes on for

- Variables declared in the for loop's initialization...

```
for (int i = 0; i < 10; i++) {
     cout <<  i << endl;
}
cout << i << endl;                //..
```

Exercise

- Convert this into a for loop:

```
int i = 0;
while (i < 10) {
    cout << i << endl;
    i++;
}
```

- Convert this into a while loop:

```
for (int j = 99; j > 0; j--) {
    cout << “# bottles: “
         << j << endl;
}
```

break and continue

- You can control loop execution inside the loop
  - break:..
  - continue:..
    - Will re-evaluate the condition, and keep looping.

```
int sum = 0;
while (true) {
    int val = 0;
    cout << "Enter a value: ";
    cin >> val;
    if (val < 0) {
        break;
    }
    sum += val;
}
```

- Don’t use these often
  - They can complicate how the loops execute.
  - Can be useful for handling error conditions.

Practice Questions

- Write the following in both a for and while loop:
  - Print all multiples of 5 from 100 down to 0.

- Write a function which:
  - Accepts two numbers, m and n.
  - Returns the sum of all numbers between m and n (you may assume m <= n)
  - Use a for-loop.

Summary

- Loops run code multiple times
  - while loop
    Indefinite loop: good for running while a condition is true.
  - for loop

    Definite loop: good for running a fixed number of times.

- Use either a while or for loop
  - for loops put all looping logic on one line.
  - while loops give you more customized control.