The C in C++: Basic Syntax

12 January 2009
CMPT166
Dr. Sean Ho
Trinity Western University

TRINITY WESTERN UNIVERSITY

Control/flow constructs

- Compound statement:
  - **{ *statement*; *statement*; .... }**
- if / else:
  - **if (*test*) *statement***
- while, do/while:
  - **while (*test*) *statement***
  - **do *statement* while (*test*)**
- break, continue (inside any loop):
  - **while (true) {**
    - **If (want_to_quit) break;**
  - **}**

The “dangling else” problem

```
if (cond1)
    if (cond2)
        statement1;
else
    statement2;
```

- Which if is the else attached to?
- Solution: always use braces

```
if (cond1) {
    if (cond2) {
        statement1;
    }
} else {
    statement2;
}
```

For loops in C++

- For loops in C/C++ are different from Python
- for (*init*; *test*; *step*) *statement*
- Common usage: counting loop:
  - **for (int i=0; i<10; i++) {**
    - **cout << i << " ";**
  - **}**
- Sequence of execution:
  - *Init; test; statement; step;*
    test; statement; step;
    test; statement; step; ...

Multi-selection: switch / case

- **switch (*expr*) {**
  - **case *value*: *statement*; *break*;**
  - **case *value*: *statement*; *break*;**
  - ***etc...***
  - **default: *statement*;**
- **}**

- *Value*s must be integral (strings don't work)
- If *break* is omitted, execution falls through to next case. Useful for multiple cases:

```
switch (inputChar) {
    case "Y":
    case "y": launch_missiles();
              break;
}
```

Basic operators

- All the usual arithmetic operators still work:
  - +, -, *, / (integer div truncates), %
  - boolean operator: && (and), || (or), ! (not)
  - bitwise operators: &, |, ^ (xor), ~ (compl)
  - bitwise shift: << (left shift), >> (right)
    - cout, cin overload these operators
- Assignment operators: +=, -=, etc.
  - increment/decrement ++ / --:
    - **numApples++;**
- Ternary operator: *test ? true_expr : false_expr*

Character input/output

- We've already seen cout in HelloWorld.cpp:
  - **#include <iostream>**
  - **using namespace std;**
  - **cout << “Hello, World!” << endl;**
- Send manipulators to cout to change the formatting of the output stream:
  - cout << “in hex: ” << hex << 15 << endl;
- Read input using cin and >>:
  - **int numApples;**
  - **cout << “How many apples? ”;**
  - **cin >> numApples;**

Arrays

- Remember the limitations of C arrays:
  - statically typed, fixed size
    - `int appleBins[5] = { 10, 3, 17, 4, 0 };`
    - `appleBins[0]++;`
    - `for (int i=0; i<5; i++) {`
      - `cout << appleBins[i] << endl;`
- No safety checks if you access out-of-bounds!
- For something closer to Python's lists, see the C++ STL class vector.