An Introduction to Programming

through C++

with

Manoj Prabhakaran

Lecture 4

Revision

More Examples

Based on material developed by Prof. Abhiram G. Ranade

So far

- Control flow: sequential, `if-else` conditions, loops
- Variables, types (`int`, `char`, `bool`, `...`), operators, expressions
- Assignment, incrementing/decrementing

Today

- More examples

Incrementing


```
for (int i = 0; i <= 4; ) { // update in the body
  cout << " i == " << i;
  cout << " i++ == " << i++; // also try ++i
  cout << " i == " << i << endl;
}
```

for Example

Demo

```
cin >> noskipws;

char c; unsigned int n;
for(cin >> c; c<'0' || c>'9'; cin >> c);
for(n=0; c>='0' && c<='9'; n=n*10+(c-'0'),cin >> c);

cout << "Read number " << n << endl;
```

What does this do?

Empty body!
(Be sure to have the ; or {})

- ***Exercise:*** Remove the ; indicating the empty body of the `for` statements and execute. Explain what you observe.

`for` Example: Prime Factorisation

```
const string prompt = "Enter a non-negative number (0 to exit) : ";
unsigned int x;
for(cout << prompt, cin >> x; x!=0; cout << prompt, cin >> x ) {
   cout << "Prime factors of " << x << ": ";
   if (x == 1)
      cout << "1 has no prime factors!" << endl;
   else { // for each d, find and remove all factors of d from x
      for (int d=2; x > 1; ++d)
         for( ; x%d == 0; x /= d)
            cout << d << " ";
      cout << endl;
   }
}
```

expr1, *expr2* evaluates both (and takes on the value of the second one)

empty is OK

`for` Example: Prime Factorisation

```
const string prompt = "Enter a non-negative number (0 to exit) : ";
unsigned int x;
for(cout << prompt, cin >> x; x!=0; cout << prompt, cin >> x ) {
   cout << "Prime factors of " << x << ": ";
   if (x == 1)
      cout << "1 has no prime factors!" << endl;
   else { // for each d, find and remove all factors of d from x
      for (int d=2; x > 1; ++d)
         for( ; x%d == 0; x /= d, cout << d << " ") ;
      cout << endl;
   }
```

expr1, *expr2* evaluates both (and takes on the value of the second one)

empty is OK

This is a valid expression!

empty body!

Valid but obscure! Use update expression for "updates" only

for Example: Inscribed Squares


```
int nsqr = 4; // number of squares to draw
float side = 400, step = 5;
for (int k = 0; k < nsqr; ++k, side /= sqrt(2)) {
   for (int it = 1, nsteps = side/step; it <= nsteps*4; ++it) {
      // count from 1: no turn at the beginning, turn at the end
      if (it % 2 == 0) penDown(); else penUp();
      forward(side/nsteps);  // approximately step long
      if (it % nsteps == 0) right(90);
   }
   // prepare for the next inner square
   penUp(); forward(side/2); right(45); penDown();
}
```

Two Turtles in a Box

The Plan

```
//some constants (box size, step size)

// turtles' positions, orientations (0,90,180, or 270 degrees)
int x=-100, y=0, deg=0; // active turtle
int xp=100, yp=0, degp=180; // inactive turtle
// move from origin to active turtle's position
while (true) {
  // read one command and handle it; break on quit command
  // move from active turtle to inactive turtle
  // swap active and inactive turtles
}
```

Two Turtles in a Box (ctd.)

```
// read one command and handle it; break on quit command
char input; cin >> input;
if (input == 'f') {
  int dx = deg==0? step : (deg==180 ? -step : 0);
  int dy = deg==90? step : (deg==270 ? -step : 0);
  if (abs(x+dx) >= limit || abs(y+dy) >= limit)
    cout << "Can't hit the box!" << endl;
  else if (x+dx==xp && y+dy==yp)
    cout << "Can't collide!" << endl;
  else { forward(step); x += dx; y += dy;  }
} else if (input == 'l') { left(90); deg = (deg + 90) % 360;
} else if (input == 'r') { left(270); deg = (deg + 270) % 360;
} else if (input == 'q') break;
```

Two Turtles in a Box (ctd.)


```
// move from origin to active turtle's position
penUp(); forward(x); left(90); forward(y); left(deg-90); penDown();
...
// move from active turtle to inactive turtle
penUp();
left(-deg); forward(xp-x); left(90); forward(yp-y); left(degp-90);
penDown();

// swap active and inactive turtles
int tmp;
tmp = x; x = xp; xp = tmp;
tmp = y; y = yp; yp = tmp;
tmp = deg; deg = degp; degp = tmp;
```

A Squiggle


```
int toothwidth = 20, toothheight = 40;

// we start parallel to the direction of drawing
for(int i=0, T=90; i<10; i++, T = -T) {
   forward(toothwidth);
   left(T); // turn perpendicular, alternating left/right
   forward(toothheight);
   right(T);   // turn to the original parallel direction
}
```

Two Squiggly Turtles in a Box


```
boolean squiggle=false, squigglep = false;
```

```
  if (!squiggle) forward(step);
  else
    for(int i=0, T=90; i<10; i++, T = -T) {
      forward(step/10); left(T); forward(step/5); right(T);
    }
```

```
} else if (input == 's') { // hidden feature!
    squiggle = !squiggle; // toggle squiggle for current turtle
    cout << "Squiggle turned " << (squiggle? "on" : "off")
         << endl;
    continue; // don't swap turtles
```