

Pascal vs C++ (Continue)

Lecture 2

Professor Peter Cheung
EEE, Imperial College

- In this lecture*,
 - We will continue to highlight the basic feature of C++
 - We will focus on the part of C++ that is derived from C
 - We will compare this with the syntax found in Pascal

* Chapters 2 of Savitch

Pascal vs C++ (16 - Strings)

Pascal	C++
No such thing in standard Pascal, but many environments provide a String type and operators that handle strings.	Many standard library functions are provided for handling null-terminated strings. Null-terminated strings use a null (' <code>\0</code> ', zero value) character to mark the end of the string. Literal strings (e.g. "Hello") are null-terminated, so the actual number of bytes required for storage is the number of characters plus one: <code>char hello[6] = "Hello";</code> A String class is provided in many environments, but it is not completely standardized, yet.

Pascal vs C++ (17 - if-else Statements)

Pascal	C++
<pre> if <condition> then <statement>; if <condition> then <statement1> {semicolon not allowed} else <statement2>; if <condition> then begin <statement11>; <statement12>; end else begin <statement21>; <statement22>; end;</pre>	<pre> if (<condition>) <statement>; if (<condition>) <statement1>; // semicolon required else <statement2>; if (<condition>) { <statement11>; <statement12>; } else { <statement21>; <statement22>; }</pre>

Pascal vs C++ (18 - Loops)

Pascal
<pre> { while } while <condition> do begin <statement>; ... end; { for } for i := n1 to n2 do begin ... { the step is 1 } end; for i := n2 downto n1 do begin ... {the step is -1 } end; for ch := ltr1 to ltr2 do ... { repeat -- until } repeat <statement>; ... until <condition>;</pre> repeat-until keeps iterating as long as <condition> remains <u>false</u>; quits when condition becomes <u>true</u>.

Pascal vs C++ (18 - For loops)

C++

```
// for
for ( <initialization>; <condition>; <increment> ) {
    ...
}

// <initialization>, <condition> and <increment> are arbitrary
// statements. For example, a common idiom similar to Pascal's
// for i := 0 to n-1 do

for (i = 0; i < n; i++) {
    ...
}
```

Pascal vs C++ (18 - do/while loop)

C++

```
// while
while ( <condition> ) {
    <statement>;
    ...
}

// do -- while
do {
    <statement>;
    ...
} while ( <condition>;
```

do-while keeps iterating as long as <condition> remains true; quits when condition becomes false.

Pascal vs C++ (19 - break & continue)

Pascal

No such thing.

C++

break is used within a loop to quit early. **continue** quits the current iteration and sends control to the new iteration.
Example:

```
for (i = 0; i < n; i++) {
    if (a[i] < 0) break; // Quit for loop
    if (a[i] == 0) continue; // Continue with the next i
    product *= a[i]; // If we get here, a[i] is > 0
    ...
}
```

Pascal vs C++ (20 - case/switch)

Pascal

```
case <expression> of
    <const1>:
        <statement1>;
    ...
    <constN>:
        <statementN>;
end;
```

Examples:

```
case ch of
    'A': Add();
    'D': Delete();
    'M': Modify();
    'Q': ; { do nothing }
end;
```

```
case age >= 65 of
    true: ...;
    false: ...;
end;
```

C++

```
switch ( <expression> ) {
    case <const1>:
        <statement1>; break;
    ...
    case <constN>:
        <statementN>; break;
    default: // optional
        <dfltstatement>; break;
}
```

Examples:

```
switch (ch) {
    case 'A':
        Add();
        break;
    case 'D':
        Delete();
        break;
    case 'M':
        Modify();
        break;
    case 'Q':
        break; // Do nothing
}
```

Pascal vs C++ (21 - Input/Output)

Pascal

```
write (x, y, ...);
writeln (x, y);
writeln;
read (x, y, ...);
readln (x, y);
readln;
write (x : width : decimals);
writeln ('$ ', amt : 6 : 2);    { e.g. $ 19.00 }
```

To read a line of text:

```
var
  str : packed array [1..80] of char;
  i : integer;
  ...
{ Read to the end of the line,
  but at most 80 chars: }
i := 1;
while (not eoln) and (i <= 80) do begin
  read(str[i]);
  i := i + 1;
end;
{ Throw away the rest of the line, if any: }
readln;
...
```

Pascal vs C++ (21 - Input/Output)

C++

```
#include <iostream.h>
...
cout << x << ' ' << y << ...;
cout << x << ' ' << y << endl;
cout << endl;
cin >> x >> y >> ...;
cin >> x >> y; cin.ignore(80, '\n');
cin.ignore(80, '\n');
```

To read a line of text:

```
char str[81];
cin.getline(str, 81);
// Reads to the end of the line or
// until you get 80 chars, whichever
// happens first. Appends a null char at the end.

cin.get(str, 81).ignore(1000, '\n');
// Reads to the end of the line,
// but at most 80 chars. Appends null.
// Throws away remaining chars on the line, if any.
```

Pascal vs C++ (22 - File Copy)

Pascal

```
program CopyFile (input, output);
var
  ch : char;
  infile, outfile : text;
begin
  assign (infile, 'INPUT.TXT');
  assign (outfile, 'OUTPUT.TXT');
  reset (infile); { open for reading }
  rewrite (outfile); { open for writing }
  while not eof(infile) do begin
    read (infile, ch);
    write (outfile, ch);
  end;
  close (infile);
  close (outfile);
end.
```

C++

```
// COPYFILE.CPP
#include <fstream.h>
int main()
{
  char ch;
  ifstream infile("INPUT.TXT");
  ofstream outfile("OUTPUT.TXT");

  while (infile.get(ch))
    outfile.put(ch);
  // Files are closed automatically
  // when infile, outfile variables
  // go out of scope.

  return 0;
}
```

Pascal vs C++ (An Example - BMI)

Pascal

```
Program CheckWeight (input, output);
{
  This program prompts the user for his weight in pounds and height in inches,
  computes the Body Mass Index (BMI) and displays "Underweight" if BMI is
  less than 18, "Normal" if BMI is between 18 and 25, and "Overweight" if BMI
  is greater than 25. BMI is defined as weight, in kilograms, divided over
  the squared height, in meters.
}
const
  kgInPound = 0.4536;
  metersInInch = 0.0254;
var
  weight,
  height : real;
  BMI : integer;
function BodyMassIndex(weight, height : real) : integer;
{ Takes weight in kilograms and height in meters.
  Returns the value of BMI rounded to the nearest integer. }
var
  bmIndex : real;
begin
  bmIndex := weight / (height * height);
  BodyMassIndex := round(bmIndex);
end;
```

Pascal vs C++ (An Example - BMI)

Pascal

```
begin { main program }
  write ('Enter your height in inches ==> ');
  readln (height);
  write ('Enter your weight in pounds ==> ');
  readln (weight);
  weight := weight * kgInPound;
  height := height * metersInInch;
  BMI := BodyMassIndex(weight, height);
  writeln ('Your BMI = ', BMI);
  if BMI < 18 then
    writeln ('Underweight')
  else if BMI <= 25 then
    writeln ('Normal')
  else
    writeln ('Overweight');
end.
```

PYKC 25 Jan 2006

EE2/ISE1 Algorithms & Data Structures

Pascal vs C++ (An Example - BMI)

C++

```
/*
  BMI.CPP
  This program prompts the user for his weight in pounds and height
  in inches, computes the Body Mass Index (BMI) and displays
  "Underweight" if BMI is less than 18, "Normal" if BMI is between 18
  and 25, and "Overweight" if BMI is greater than 25. BMI is defined
  as weight, in kilograms, divided over the squared height, in meters.
  */
#include <iostream.h>

int BodyMassIndex(double weight, double height)
// Takes weight in kilograms and height in meters.
// Returns the value of BMI rounded to the nearest integer.
{
  double bmIndex;
  bmIndex = weight / (height * height);
  return int(bmIndex + .5); // round to the nearest integer
}
```

PYKC 25 Jan 2006

EE2/ISE1 Algorithms & Data Structures

Pascal vs C++ (An Example - BMI)

C++

```
int main()
{
  const double kgInPound = 0.4536, metersInInch = 0.0254;
  double weight, height;
  int BMI;
  cout << "Enter your height in inches ==> ";
  cin >> height;
  cout << "Enter your weight in pounds ==> ";
  cin >> weight;
  weight = weight * kgInPound; // or: weight *= kgInPounds;
  height = height * metersInInch; // or: height *= metersInInch;
  BMI = BodyMassIndex(weight, height);
  cout << "Your BMI = " << BMI << endl;
  if (BMI < 18)
    cout << "Underweight" << endl;
  else if (BMI <= 25)
    cout << "Normal" << endl;
  else
    cout << "Overweight" << endl;
  return 0;
}
```

P

What to do before the next lecture?

- Complete Exercise A (non-assessed) – See separate sheet
- Complete Lesson 5 & Lesson 6 of the following C++ Tutorial on the web:

<http://www.functionx.com/cppbcb/Lesson05.htm>

<http://www.functionx.com/cppbcb/Lesson06.htm>

- Alternatively, read Chapter 2 of Savitch

PYKC 25 Jan 2006

EE2/ISE1 Algorithms & Data Structures