Course Overview

and Rough C++ Guide

Professor Peter Cheung EEE, Imperial College

- In this lecture,
 - we take an overview of the course, and
 - briefly review the C++ programming language.
- The rough C++ guide is not very complete. You should use a suitable book as a proper reference for the C++ language.

PYKC Jan 2006

EE2/ISE1 Algorithms & Data Structures

1/3

1/1

Recommended Texts

• You could get away with nothing. But a good C++ textbook would help, such as:

Problem Solving with C++, 5th Edition, Walter Savitch, Addison Wesley, ISBN: 0321269756, 2004 (£40.84).

• You can also get a full C++ Tutorial free on the web at:

http://www.functionx.com/cppbcb/

• If you are serious about learning really how to programme well, then the following book (language independent) is a must:

Code Complete, Steve McConnell, Microsoft Press, ISBN: 0735619670 (£25.84).



- The aim of this course is to familiarise you with a number of *principles, concepts* and *techniques* from computer science.
- These principles, concepts and techniques are general purpose. They apply
 - whatever programming language you use (Java, C++, Pascal), and
 - whatever computer you're using (a PC, a Unix system, a Mac).
- Once you know basic engineering principles, concepts and techniques you'll find it easy to pick up new programming languages, new operating systems, etc.

PYKC Jan 2006

EE2/ISE1 Algorithms & Data Structures

Principles,

Concepts & Techniques

- The *principles* you will learn include things like how to construct code that's easy to read, understand and modify.
- The *concepts* you will learn about include things like abstract data types, object-oriented programming, and so on.
- The *techniques* you will learn about include
 - how to build *data structures*, like lists and trees, and
 - *algorithms* for certain common tasks, like lookup and sorting.

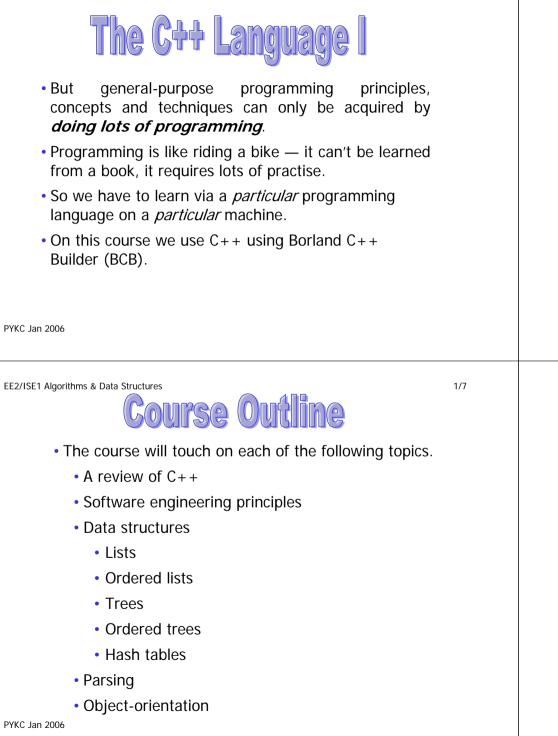
But

PYKC Jan 2006

PYKC Jan 2006

1/6

1/8



The Ct	+ Language I
	is an object oriented varia

- The C++ language is an object-oriented variant of the C language.
 - "Object-oriented" means that code and data can be bundled together into a single entity called an *object*. Classes of objects are organised into *hierarchies*. More on this later in the course.
- But Borland C++ Builder (BCB) is not just a programming language. It's a whole development environment (IDE). It includes an *editor* and a *debugger*, as well as a compiler.
- BCB is also a *visual* development environment. It supplies lots of support for building the graphical user interface (GUI) of a program. It contains a Visual C++ Library (VCL)
- which helps you to write visual programmes.

EE2/ISE1 Algorithms & Data Structures

EE2/ISE1 Algorithms & Data Structures



- C++ is a programming language.
- It is a
 - block structured.
 - strongly typed,
 - procedural language.
- It's a lot like C or Pascal or Java, but arguably more flexible and more widely used than others.

Programming Languages

- "Block structured" means that the flow of control in a program is strictly mediated by a small number of constructs, namely
 - sequences of statements,
 - conditional statements (e.g. if (<condition>) then
 {<statements>} else {<statement>}), and
 - loops (e.g. while condition {<statement>}).
- "Strongly typed" means that the programmer has to declare the *type* of each variable used in the program, ie: whether it is an integer, a string, an array, or whatever.
- "Procedural" means that the program specifies a series of instructions and the order in which they are to be carried out. In a *declarative* language, by contrast, the programmer only describes the meanings of things — functions or predicates — and leaves it to the computer to work out the details of how computations are to be performed. Prolog is an example of a declarative language.

PIKC Jan 2006

```
EE2/ISE1 Algorithms & Data Structures
```

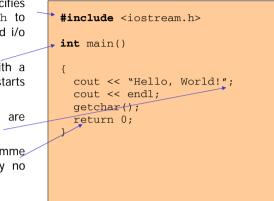
1/11

Pascal vs C++ (1 - Comments)

Pascal	C++
Comments are placed between braces: { This is a comment } The older notation: (* This is a comment *)	<pre>Comments are placed between /* and */ /* This is a comment */ The second method: a comment is placed after //. Then it extends to the end of the line: x = x+1; // A comment to the end of the line.</pre>

EE2/ISE1 Algorithms & Data Structures

- Here is a skeleton program in C++. (This is a console program. It doesn't use the fancy user interface features of BCB.)
- This include statement specifies the header file iostream.h to be used. It defines standard i/o cin and cout.
- All C++ programs begin with a line of this form. It always starts with main().
- All C++ statements ar terminated by a semicolon.
- When exit, the main programme returns a value 0 to signify no error has occurred.



PYKC Jan 2006

EE2/ISE1 Algorithms & Data Structures

Pascal vs C++ (2 - upper/lower case)

Pascal	C++
PASCAL IS CASE-BLIND.	C++ is case-sensitive.

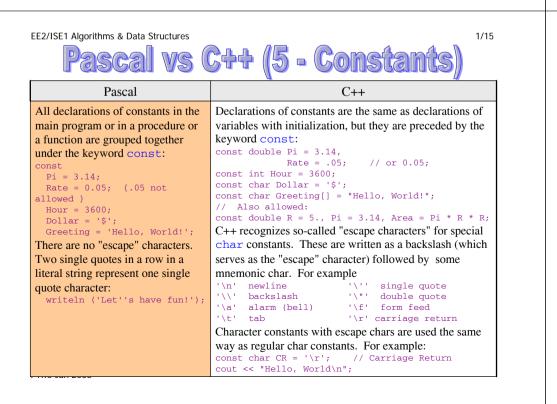
Pascal	C++
Names can use letters and digits but must begin with a letter, e.g.:	Names can use letters, digits and the underscore character, but must begin with a letter or the underscore, e.g.:
amount, x1, str3a	amount, x1_, _str3a

1/12

Pascal vs C++ (3 - Block & Semicolons)

Pascal	C++
A compound statement is placed between begin and end:	A compound statement is placed between braces:
<pre>begin <statement1> ; <statement2> end;</statement2></statement1></pre>	<pre>{ <statement1> ; <statement2> ; }</statement2></statement1></pre>
Semicolon is optional before end and is usually required after end, unless followed by another end.	Semicolon is required before the closing brace, and usually omitted after it.

PYKC Jan 2006



Pascal vs C++ (4 - Built-in & Enumerated Data Types)

Pascal		(C++
char integer real boolean	char int float	5	
	prece douk numb	ded by the un ble is a doubler. is in the prod	t, and long may be nsigned keyword. le-precision real cess of becoming
Pascal			C++
type Color = (Red, Green, Blue);		enum Color	{Red, Green, Blue};
'KC Jan 2006			

EE2/ISE1 Algorithms & Data Structures 1/16 Pascal vs C++ (6 - Variables) Pascal C++ All declarations of variables in the Declarations of variables (or constants) may main program or in a procedure or a be placed more or less anywhere in the code, function are grouped together under before they are used. Beginners are advised the keyword **var**: to place them at the top of main() or at the top of a function to avoid complications with SomeProcedure (...); the scope rules. Global variables, declared outside any function (and outside main()), var are allowed, but should be avoided. Values r : real; of variables may be initialized to constants or i, j : integer; star : char; previously defined variables or expressions: match : boolean; . . . SomeFunction (...) begin . . . double r = 5.;end; int i = 0, j = i+1;char star = '*'; No initialization is allowed in . . . declarations. PYKC Jan 2006

EE2/ISE1 Algorithms & Data Structures 1/17 Pascal vs C++ (7 - Arrays)	
Pascal	C++
<pre>var str : packed array [180] of char; grid : array [132, 125] of integer; The packed keyword is recommended for an array of characters to save space. The range of subscripts can start from any number, but usually starts from 1. Here str[1] refers to the first element of the array str. Pascal compilers normally report an error if a subscript value is out of range.</pre>	<pre>char str[80]; int grid[32][25]; The subscript for the first element of the array is 0. Here str[0] refers to the first element of the array str and str[79] to the last element. C++ compilers <u>do not</u> verify that a subscript value is within the legal range. Arrays can be initialized in declarations. For example: int fibonacciNumbers[6] = {1,1,2,3,5,8}; char phrase[80] = "Hello, World!";</pre>

PYKC Jan 2006



Pascal

Procedures and functions take arguments of specified types. Procedures do not explicitly return a value. Functions return a value of the specified type.

Pascal vs C++ (8 - Type Definition)



The type keyword is used to define enumerated and *subrange* types, array types, and records:

type DigitType = 0..9; { Subrange type } ColorType = (Red, Green, Blue); { Enumerated type } WordType = packed array [1..30] of char; { Array type }



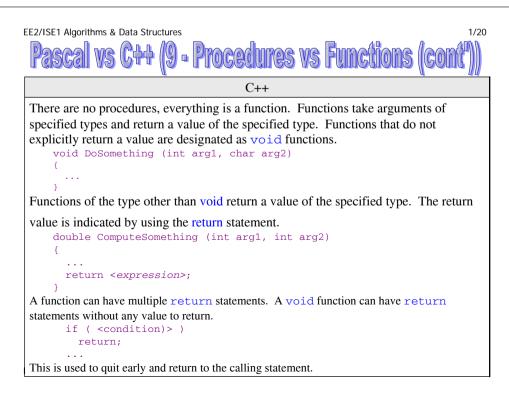
The typedef keyword is used to define aliases for built-in (and, if desired, user-defined) types:

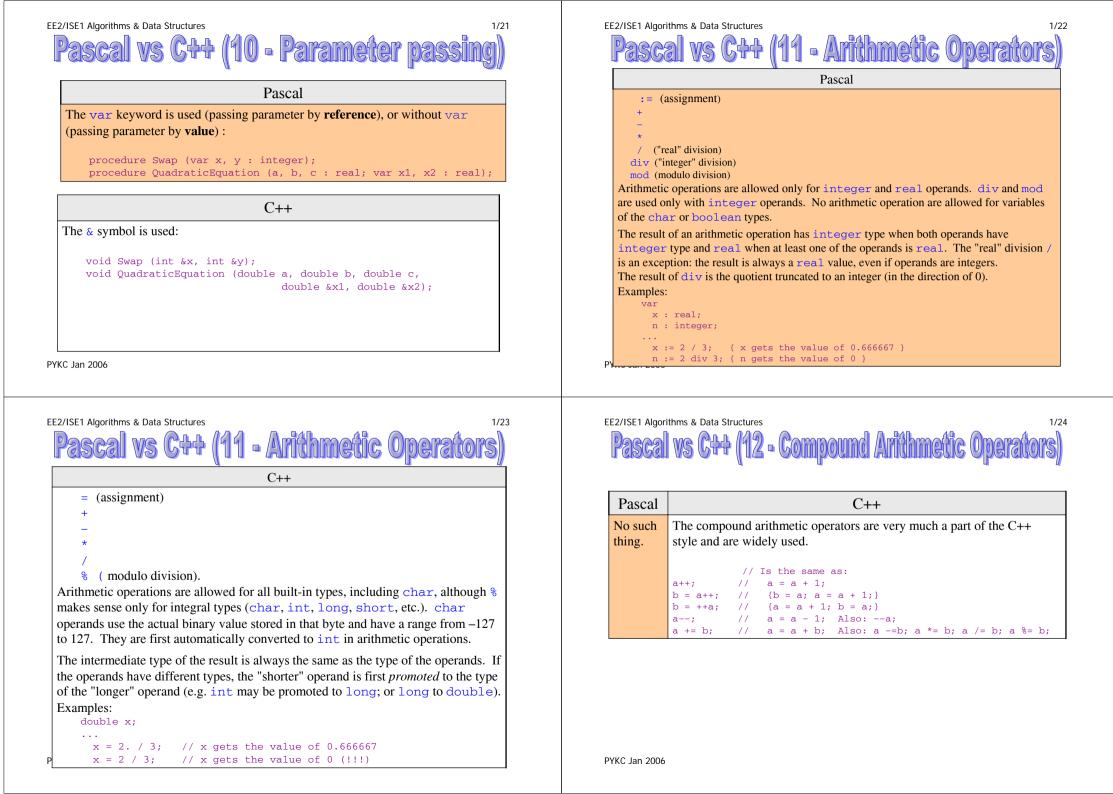
typedef unsigned char BYTE; typedef double MONEY; typedef int BOARD[8][8]; // Used later in declarations as:

- // BYTE pixel;
- // MONEY price = 9.95;
- // BOARD board;

PYKC Jan 2006

1/19



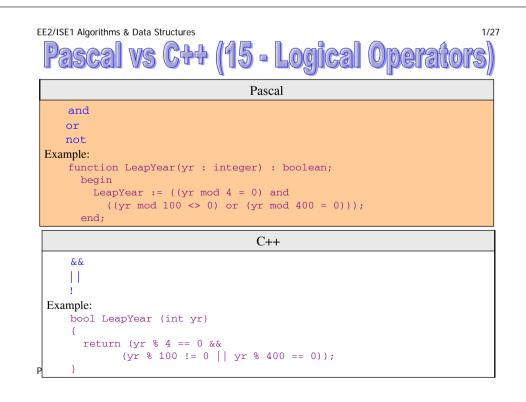


EE2/ISE1 Algorithms & Data Structures

1/25 Pascal vs C++ (13 - Built-in Maths Functions)

Pascal	C++	
Built-in functions:	Standard library functions (require #include	
	<math.h>):</math.h>	
abs(x)		
sqrt(x)	<pre>int abs(int x); double fabs(double x);</pre>	
sin(x)	<pre>double sqrt(double x);</pre>	
cos(x)	<pre>double sin(double x);</pre>	
exp(x)	<pre>double cos(double x);</pre>	
ln(x)	<pre>double exp(double x);</pre>	
sqr(x)	<pre>double log(double x); // Natural logarithm</pre>	
arctan(x)	<pre>double pow(double base, double exponent);</pre>	
	<pre>double atan(double x);</pre>	

PYKC Jan 2006



EE2/ISE1 Algorithms & Data Structures

1/26 Pascal vs C++ (14 - Boolean variables & values)

Pascal	C++	
Has built-in boolean type and constants true	Any integer non-zero value is treated as "true," and zero as "false." bool type is in the process of becoming standard.	
and false.	If not supported by their compiler, programmers may use their own definition. For example:	
	typedef int bool; #define false 0 #define true 1	
Descel	Chi	

Pascal	C++
The result of relational operators	The result has the type bool and has the value
has the type boolean.	false or true:
=	==
<>	!=
<	<
<=	<=
>	>
	>=
>=	

EE2/ISE1 Algorithms & Data Structures

1/28

hat to do before the next lecture?

- Use Level 3 computer lab or install a copy of BCB on your own machine.
- Complete Lesson 1 & Lesson 2 of the following C++ Tutorial on the web:

http://www.functionx.com/cppbcb/Lesson02.htm