

A SHORT COURSE ON C++

Dr. Johnson

School of Mathematics

Semester 1 2008

OUTLINE

- 1 INTRODUCTION TO C++
 - Object-Orientated Programming
 - Syntax
 - Handling Data and Variables
 - Input/Output
- 2 FLOW CONTROL AND FUNCTIONS
 - If Else
 - Looping
 - Functions
 - Cmath Library
 - Prototyping

- A structured language can *hide* information from the rest of the program.
- Structuring code and data allows
 - easy upgrades
 - many programmers to work on a large project
- Object-oriented programming imposes a high level of structure
- Problems are broken down into subproblems, and then into self-contained units called objects
- Common traits of object-oriented languages are:
 - encapsulation
 - polymorphism
 - inheritance

USING OBJECTS

1 Encapsulation:

functions and data inside an object have restricted access.

2 Polymorphism:

represents the concept of “one interface, multiple method”. The same interface can be used to do different things for different objects: i.e. define + to add numbers, but perform string concatenation on characters and strings, ‘a’ + ‘b’ = ‘ab’.

3 Inheritance:

allows one object to acquire the properties of another. An example would be to define a generic object “car” that has a steering wheel, four wheels and an engine. The new object “sports car” inherits all these properties and adds a sun roof, go-faster stripes and a huge stereo.

WRITING C++

The key elements of C/C++ syntax are:

- Semicolon used to mark end of statements
- Case is important
- Totally free form, lines and names can be as long as you like!
- Comments take the form `/* C style comment */` or `// C++ style comment`
- Code blocks are surrounded by braces `{ }`

A VERY SIMPLE C++ CODE

- The following is a C++ program.

```
main()  
{  
}  
}
```

- There are no commands to execute.
- If we save it in the file “simple_prog.cc”,
- we can compile and run it with the commands:

```
> c++ simple_prog.cc  
> ./a.out
```

This is a sample, click download link to get the full Tutorial

CLICK BELOW

