This document provides an introduction to computing and the C++ programming language. It will help you teach yourself to write, compile, execute and test simple computer programs in C++ and describes some of the computing exercises to be completed in the IB course.

Please read the whole of this tutorial guide as soon as possible. Pages 1–43 must be read before the first laboratory session. Pages 32–48 should be read before the second laboratory session. Pages 49–63 should be read before the third laboratory session. The rest of the tutorial guide should be read before the second term.

For the latest information about the IB course, please see:

- the course wiki (which requires Raven access):
  
  https://wiki.csx.cam.ac.uk/cphysics/

  – also reachable via http://tinyurl.com/32tw9m

- the course homepage:

  http://www.inference.phy.cam.ac.uk/teaching/comput/C++/

  – also reachable via http://tinyurl.com/2uffap

You are advised to attend the laboratory sessions with expert demonstrators, which will take place in the PWF at the Cavendish Laboratory (Room 173, Bragg building, next to the 1A labs) at the following times in weeks 5, 6, 7, 8 of Michaelmas term and weeks 1–3 of Lent term.

<table>
<thead>
<tr>
<th>Time</th>
<th>Monday</th>
<th>Tuesday</th>
<th>Wednesday</th>
<th>Friday</th>
</tr>
</thead>
<tbody>
<tr>
<td>11am–1pm</td>
<td></td>
<td>11am–1pm</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2–4pm</td>
<td>2–4pm</td>
<td>2–4pm</td>
<td>2–4pm</td>
<td>2–4pm</td>
</tr>
</tbody>
</table>

Bring your PWF password!
# Contents

1  Introduction 17  
2  Editing, compiling and executing a simple program 17  
3  Variables and constants 26  
4  Assignment of variables 27  
5  Simple input and output 31  
6  Control Statements 32  
7  Functions 49  
8  Math library and system library built-in functions 57  
9  Arrays 64  
10  Reading and writing to files 72  
11  Arrays done better 73  
12  Modularizing 75  
13  Structures and packages 79  
14  Formatted output 83  
A  An introduction to object-oriented programming and classes 96  
B  Further reading 98
A. Aims of the Computing Course

This guide provides a tutorial introduction to computing and the C++ computer programming language. It will help you teach yourself to write, compile, execute and test simple computer programs in C++ which read data from the keyboard, perform some computations on that data, and print out the results; and how to display the results graphically using gnuplot.

By the end of the course, you should be able to use computing to help you understand physics better.

C. Organisation

There are 4 two-hour laboratory sessions scheduled for the Michaelmas term computing course, and 3 sessions for the Lent term. You should work through the examples and exercises described in this document. Each session consists of a tutorial (which must be read before the start of the laboratory session) and interleaved computing exercises.

Please allow two hours preparation time per week.

The computing exercises are placed inside a box like this one.

To benefit most from the laboratory session and from the demonstrators’ help you must read the tutorial sections before the start of each session.

C. Assessment

In this course, your progress will be self-assessed.

The self-assessment process is based on a desire to treat you as adults, and on the assumption that you are interested in learning, capable of choosing goals for yourself, and capable of self-evaluation. I expect everyone to achieve full marks.

Programming is an essential skill, just like graph-sketching. If you don’t know how to sketch graphs, what do you do? – practice, read books, talk to supervisors and colleagues, practice some more. As second-year physicists, graph sketching should be second nature, and you should use it as a tool in all physics problems you study. Similarly, programming is a valuable tool for thinking, exploring, investigating, and understanding almost any scientific or technical topic. You will find that programming can help you with your 2nd year, 3rd year, and 4th year courses. This course exists to get you up to speed on programming, for the benefit of all your courses.

Many skills are best acquired gradually. Programming is one such skill, so this course is spread over two terms.

To ensure that you are making steady progress we will set a sequence of tasks and targets. We strongly encourage you to do these tasks at the recommended time, because that’s when the expert demonstrators are there to help you. To allow for unexpected complications, the final deadline for assessment of each piece of work will be 9 days after the recommended completion time.
End of ebook preview
Download the full PDF tutorial from the link below:

Click Here