

# Natural Sciences, Part IB, Introduction to Computing

## C++ Programming Tutorial and Instructions for Practical Sessions

Christopher Lester  
Department of Physics

(based on earlier versions by David MacKay, Roberto Cipolla and Tim Love)

November 4, 2013

- The course homepage is : <http://www.hep.phy.cam.ac.uk/lester/c++2013/>
- There is also a course wiki at <https://wiki.csx.cam.ac.uk/cphysics/>

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 6–25 must be read *before* the first laboratory session.

Pages 25–48 should be read before the second and third laboratory session.

Pages 48–64 should be read before the fourth laboratory session.

The rest of the tutorial guide should be read before the remaining two sessions.

### Laboratory Sessions:

You are advised to attend the hands-on lab sessions. These sessions run from Thursday 17th October to Wednesday 4th December 2013 inclusive, and take place in the Cavendish in the same building as practical classes. Demonstrators will be on-hand each day between 2pm and 4pm to help you and answer your questions, though the room itself can be used by students at any time it is open (see notices). Attendance at one session per week should be sufficient to complete the course.

**Bring your PWF password to the laboratory sessions or you will be unable to log in!**

# Contents

<b>1</b>	<b>Introduction</b>	<b>6</b>
1.1	Aims of the Computing Course . . . . .	6
1.2	Organisation of Laboratory Sessions . . . . .	6
1.3	The Six Pieces Of Submitted Work (“SESSIONS”) . . . . .	6
1.4	Why self assessment? . . . . .	7
1.5	Assessment deadlines . . . . .	8
1.6	Collaboration . . . . .	8
1.7	Copying . . . . .	9
1.8	Feedback . . . . .	9
<b>2</b>	<b>Introduction</b>	<b>10</b>
2.1	What is a computer program? . . . . .	10
2.2	The C++ Programming Language . . . . .	10
<b>3</b>	<b>Editing, compiling and executing a simple program</b>	<b>10</b>
3.1	A simple C++ program to add two numbers . . . . .	10
3.2	Overview of program structure and syntax . . . . .	11
3.3	The development environment and the development cycle . . . . .	13
<b>4</b>	<b>INSTRUCTIONS FOR PRACTICAL SESSION 1</b>	<b>14</b>
4.1	Objectives . . . . .	14
4.2	Getting started . . . . .	14
4.3	Computing exercise – editing, compiling and executing a simple program that adds numbers . . . . .	17
4.4	Exploring compiler error messages . . . . .	18
4.5	Compiler warnings . . . . .	19
4.6	Makefiles . . . . .	19
4.7	Practice . . . . .	19
4.8	Assessment . . . . .	21
4.9	Instructions for submitting your self-assessed work . . . . .	21
<b>5</b>	<b>Variables and constants</b>	<b>25</b>
5.1	Variable types . . . . .	25
5.2	Declaration of a variable . . . . .	25
5.3	Storage of variables in computer memory . . . . .	26
<b>6</b>	<b>Assignment of variables</b>	<b>26</b>
6.1	Assignment statements . . . . .	26
6.2	Arithmetic expressions . . . . .	27
6.3	Precedence and nesting parentheses . . . . .	28
6.4	Initialisation of variables . . . . .	28
6.5	Expressions with mixed variable types . . . . .	28
6.6	Declaration and initialisation of symbolic constants . . . . .	29
<b>7</b>	<b>Simple input and output</b>	<b>29</b>
7.1	Printing to the screen using output stream . . . . .	30
7.2	Input of data from the keyboard using input stream . . . . .	30

<b>8</b>	<b>Control Statements</b>	<b>31</b>
8.1	Boolean expressions and relational operators . . . . .	31
8.2	Compound boolean expressions using logical operators . . . . .	31
8.3	The IF selection control statement . . . . .	31
8.4	The IF/ELSE selection control statement . . . . .	32
8.5	ELSE IF multiple selection statement . . . . .	33
8.6	SWITCH multiple selection statement . . . . .	35
8.7	The WHILE repetition control statement . . . . .	37
8.8	Increment and decrement operators . . . . .	38
8.9	The FOR repetition control statement . . . . .	38
8.10	The DO... WHILE repetition control statement . . . . .	40
<b>9</b>	<b>INSTRUCTIONS FOR PRACTICAL SESSION 2</b>	<b>41</b>
9.1	Objectives . . . . .	41
9.2	Computing Exercises . . . . .	41
9.3	Tips . . . . .	42
9.4	Assessment . . . . .	43
<b>10</b>	<b>INSTRUCTIONS FOR PRACTICAL SESSION 3</b>	<b>44</b>
10.1	Objectives . . . . .	44
10.2	Computing Exercises . . . . .	44
10.3	Assessment . . . . .	47
<b>11</b>	<b>Functions</b>	<b>49</b>
11.1	Function definition . . . . .	49
11.2	Example of function definition, declaration and call . . . . .	50
11.3	Function header and body . . . . .	52
11.4	Function declaration . . . . .	53
11.5	Function call and execution . . . . .	53
11.6	Function arguments . . . . .	54
11.7	Another example . . . . .	55
11.8	Passing by value or reference . . . . .	56
<b>12</b>	<b>Math library and system library built-in functions</b>	<b>57</b>
12.1	Mathematical functions . . . . .	58
12.2	Random numbers . . . . .	58
12.3	How can I find out what library a function is in? . . . . .	59
<b>13</b>	<b>INSTRUCTIONS FOR PRACTICAL SESSION 4</b>	<b>61</b>
13.1	Objectives . . . . .	61
13.2	Computing Exercises . . . . .	61
13.3	The bisection method . . . . .	61
13.4	Notes on algorithm and implementation . . . . .	62
13.5	Assessment . . . . .	64
13.6	‘Have I done enough?’ . . . . .	65

<b>14 Arrays</b>	<b>66</b>
14.1 Declaration . . . . .	66
14.2 Array elements and indexing . . . . .	66
14.3 Assigning values to array elements . . . . .	67
14.4 Passing arrays to functions . . . . .	69
14.5 Character arrays . . . . .	69
14.6 Multi-dimensional arrays . . . . .	70
14.7 Structures . . . . .	73
14.8 Enumerated constants . . . . .	73
<b>15 Reading and writing to files</b>	<b>74</b>
<b>16 Direct allocation of arrays with “new”</b>	<b>76</b>
<b>17 Modularizing</b>	<b>77</b>
<b>18 Structures and packages</b>	<b>81</b>
<b>19 Formatted output</b>	<b>85</b>
<b>20 Notes concerning the remaining SESSIONS 5, 6 and 7.</b>	<b>86</b>
<b>21 INSTRUCTIONS FOR PRACTICAL SESSION 5 : “PLANET”</b>	<b>87</b>
21.1 Objectives . . . . .	87
21.2 Task . . . . .	87
21.3 Ideas for what to do . . . . .	88
21.4 What to hand in . . . . .	90
<b>22 INSTRUCTIONS FOR PRACTICAL SESSION 6 : “BONKERS”</b>	<b>91</b>
22.1 Objectives . . . . .	91
22.2 Task . . . . .	91
22.3 Ideas for what to do . . . . .	92
22.4 What to hand in . . . . .	93
<b>23 INSTRUCTIONS FOR PRACTICAL SESSION 7 : “RECURSION” (optional!)</b>	<b>95</b>
23.1 Objectives . . . . .	95
23.2 Recursion exercises . . . . .	98
<b>A PLANET: step by step guidance</b>	<b>99</b>
<b>B An introduction to object-oriented programming and classes</b>	<b>102</b>
<b>C Further reading</b>	<b>104</b>
<b>D Objectives of each section</b>	<b>105</b>
D.1 Session 1 . . . . .	105
D.2 Session 2 . . . . .	105
D.3 Session 3 . . . . .	105
D.4 Session 4 . . . . .	105
D.5 Session 5 : “PLANET” . . . . .	105

D.6 Session 6 : “BONKERS” . . . . .	105
D.7 Optional session 7 : “RECURSION” . . . . .	106
<b>E Thirty useful unix commands</b>	<b>107</b>

[Click here to download full PDF material](#)