# ACSE-5: Advanced Programming 2020/2021

## Getting started

This module starts with a primer on C++ programming, and will teach you how to develop and build C++ programs. Students will be taught functions and input/output handling, as well as memory handling concepts such as pointers, memory allocation and advanced data structures such as linked lists and trees.

You will be taught practical skills for analysing software for potential bugs as well as quickly locating bugs when they arise using debuggers. Additionally, you will learn how to use profiling techniques to identify performance hotspots in software and gain a practical understanding of the factors that limit the performance of software.

This will be followed by an introduction to object-oriented programming with C++ including how to define and use classes, inheritance, overloaded functions and polymorphism.

These will all be taught in the context of scientific programming and numerical methods, with relevant examples and exercises being used to illustrate the programming concepts. Regular programming homework/assignments will bring together all the skills that will be learned throughout the course and build upon the skills taught in the previous programming and numerical methods courses as part of ACSE.

## Learning outcomes:

On successful completion of this module, students will be able to:

1. Write programs to solve scientific problems using C++.
2. Analyse problems in order to solve them with algorithms and software.
3. Analyse and identify opportunities for refining computer programmes.
4. Manage contributions to group work co-ordinating and co-operating with peers.

## Get started

In preparation for your ACSE-5 lecture on “Advanced Programming” we have the following instructions/info:

- As part of our ACSE-5 we will be primarily using either Microsoft Visual Studio, an IDE (integrated development environment) for the C++ programming language (https://en.wikipedia.org/wiki/Microsoft_Visual_Studio), or Microsoft Visual Studio Code, a fully featured open-source editor (https://code.visualstudio.com/) .
- The most basic edition of Visual Studio, the Community edition (https://visualstudio.microsoft.com/vs/community/), is available free of charge. This is available for C++, for *Windows and Mac* only (and includes a compiler).
- Visual Studio Code is available for Windows, Mac and Linux and is free. Please be aware you will need to have a separate C++ compiler installed if you chose to use Visual Studio Code.

- Please install either Microsoft Visual Studio Community 2019 or Microsoft Visual Studio Code (and a C++ compiler) in your laptops.
- The installation for Visual Studio Community 2019 may take a few hours. After you download the install program for Visual Studio Community 2019 and run it, you will be presented with a screen. Select the option to install "Desktop development with C++". After installing Visual Studio Community make sure that you can create and save a file to a local folder.
- If you cannot install Visual Studio Community or Visual Studio Code for space or any other reason, please send me an email as soon as possible (apaluszn@imperial.ac.uk).
- During our first class we will make sure that everyone has access to their chosen IDE/editor we will get started on some simple IDE tasks to set us up for the term.
- We will be using the "Live Share" extension available for both Visual Studio Community and Visual Studio Code to share code collaboratively (https://visualstudio.microsoft.com/services/live-share/). Please install this if necessary before class (normally this comes pre-installed for users of Visual Studio Community)

# Reference Sites

- Google's C++ Class (https://developers.google.com/edu/c++/)
- Intro to C++ (http://www.cplusplus.com/doc/tutorial/)
- Creator of C++ Bjarne Stroustrup (http://www.stroustrup.com)
- C++ "deity" Scott Meyers (http://www.aristeia.com) See books: Effective C++, More Effective C++, Effective STL

# Reference Books

For those that love the touch of paper, we suggest:

- (*most used*) Bjarne Stroustrup: Programming -- Principles and Practice Using C++. Second Edition (2014). Addison-Wesley. 1312 pages. Recommended chapters: 2, 3, 4, 5.1-5.5, 7, 8, 9, 10, 17, 18, 19, 20, 21, 24
- Stanley Lippman, Josée Lajoie, and Barbara E. Moo: C++ Primer (updated for C++11) Fifth edition (2012). Addison Wesley. 976 pages.
- Bjarne Stroustrup: A Tour of C++. Second Edition (2018). Addison-Wesley. 240 pages.

# Summary of teaching

**There will be no lectures or practicals on Tuesday the 2nd of February.**

We have lectures from 9am-12pm each day (excluding Thursday and Friday the 4th and 5th of February), please see your calendars for an up-to-date timetable and the meetings to join.

We have two separate practical sessions, one from 8am-9am and then one from 1pm-4pm each day, which will be attended by GTAs. For the practicals, all students will be grouped into one of 10 channels under the "ACSE-20 EA" main team. GTAs and the lecturers will then pop into each channel throughout the sessions to answer any questions/talk through the work within each channel. Any useful questions/answers from each channel can then be posted to the main ACSE-5 channel so everyone can

benefit. Please add yourself to a channel of your choosing by placing your name in the spreadsheet linked below. Anyone who has not added themselves to a channel on the spreadsheet by Sunday the 17th of January will be randomly assigned to a channel.

**Spreadsheet link:**
https://docs.google.com/spreadsheets/d/1wvJza5a0WOizCFmxrGAhYZGY6GkLqKI7iO27dswilmg/edit?usp=sharing

# Teaching team

- Dr Adriana Paluszny - Senior Lecturer/Royal Society University Research Fellow - computational scientist and C++ devotee
- Dr Steven Dargaville - Post-doctoral Research Associate - numerical methods connoisseur and advanced programmer
- Cristina Saceanu - PhD Student and C++ enthusiast
- Deborah Pelacani Cruz - PhD Student and C++ enthusiast
- Jorge Avalos Patino - PhD Student and C++ enthusiast
- Giannis Nikiteas - PhD Student and C++ enthusiast
- John Walding - PhD Student and C++ enthusiast

# Assignments and Evaluation

There will be two homeworks (not marked), **one group assignment** (marked – 50% of final grade), and one individual, in-class, computer-based programming **final coursework** (marked – 50% of final grade). The homework should be completed after class during the afternoon sessions, these will not be collected nor assessed; you will receive GTA support to help during the practical sessions.

Submission of the group assignment and final coursework will be via GitHub [details will follow].

1. **Homework** 1: Climate Change Focus: How can we predict temperature changes? Released Monday 18th of January. (Looking for a plotting solution? Try downloading gnuplot)
2. **Homework** 2: Implementing a medical imaging filter. Released Thursday 21st of January.
2. **Assignment**: Group project, required to have 3 students per group. Implementing linear solvers. Released Monday 25th of January, due Sunday 7th of February @ midnight - early submissions welcome.
3. **Final Coursework**: The final piece of coursework will be released and submitted on Monday Febuary 1st in the lecture time, 9am-12pm. This is an individual piece of work and standard College plagiarism rules apply (i.e., this must be completed by yourself alone with no external help; **please note** that unfortunately students have been caught and sanctioned for plagiarism in ACSE-5 in previous years). You will be able to make use of all the code you have written for the group assignment.

## Feedback and Questions

- You can send any questions/feedback to Adriana (apaluszn@imperial.ac.uk) or Steven (s.dargaville@imperial.ac.uk) or ask any of the teaching team directly during class.

## Final checklist before class starts

- Installed Visual Studio Community or Visual Studio Code (and a C++ compiler if necessary)
- Installed the “Live Share” extension if necessary
- Assigned yourself to a channel for the practical by filling out your name in the spreadsheet