Abstracts

An experiment in embedding and distributing programming throughout an undergraduate degree
Anna Wilson University of Stirling

Ensuring adequate experience of programming and numerical methods for students in an undergraduate physics programme is always a challenge in what is a packed and varied curriculum. This challenge is perhaps even harder to meet in the Australian context, where students enroll in a Bachelor of Science and major or minor in physics (often waiting until quite late in their degree to decide on their major), and thus take substantially fewer physics modules than would be the case for a UK undergraduate.
Until the early 2000s, the Department of Physics at the Australian National University adopted the fairly common approach of offering an elective module in computational physics to third year undergraduates. By about 2005, enrollments in this module - never very high - had plummeted and so it was effectively taken off the books.
A university-wide programme restructure and an accompanying process of developing learning outcome statements at the Major and Minor level made public what we already knew: that we were failing to provide any educational experiences leading to the development of what we ourselves identified as core skills needed by contemporary physicists. This forced a re-think, leading ultimately to the experiment in introducing programming and numerical approaches as a thread embedded throughout the degree, rather than as an avoidable bolt-on. I will describe the processes we went through to plan and organize this approach, including difficulties and conflicts, and describe the current state of affairs.

Implementation of computational physics in a new Applied Physics degree
Chris Dewdney University of Portsmouth

The development and implementation of the new Physics degrees BSc/MPhys Applied Physics (started 2010) and BSc/MPhys Physics, Astrophysics and Astronomy (starting 2015) at the University of Portsmouth are discussed. The vital role played by the Higher Education Community in informing the design of the curriculum and pedagogic techniques adopted is illustrated as is the important input received form Industry and other employers. Throughout the degrees' design process it was clear that computing skills were both highly valued by employers and felt to be in deficit by recently employed graduates.

We reconsidered the vexed question of which computation environment to adopt and in the end took into account advice from our Industry Advisory Board. We adopted an approach that synchronized and integrated computation across several units at level 4 and level 5. Adopting a PBL approach we implemented a "bottom-up" strategy for developing computing skills as part of a set of tools for problem solving that integrate experiment, theory and simulation across the curriculum.

Clearly our implementation is in its early stages and has yet to be evaluated systematically. However, we believe that the approach we adopted serves to emphasize the coherence of theory, experiment and computation whilst challenging students to employ all the tools at their disposal to troubleshoot and understand their simulations and their contribution to problem solving.

Teaching physics with Python
Louise Dash University College London

At UCL we have been revising our core computing courses. Over the last two years we have introduced new Python-based courses, with the aim of not only enabling students to analyse and present their data, but also to enhance their understanding of physics by creating and manipulating their own computational models. Python provides an ideal environment for this, and in particular IPython Notebooks enable students to not only program effectively but also to consider more deeply what they are coding and why.

In this talk I will share our experiences in implementing these courses, both positive and negative, with a particular emphasis on the challenges faced in teaching programming to first year undergraduates with a wide range of backgrounds. I will also discuss our plans to formally integrate computational skills in the wider curriculum.

Visual programming: LabVIEW & donkeys in the laboratory
Rachel Edwards University of Warwick

LabVIEW is fast becoming an industry standard, and is well suited for communication with laboratory equipment. At Warwick we have devised an undergraduate laboratory which requires LabVIEW programming as an integral part of the experiments, for example for outputting numbers to an LED board or controlling one of the nodding donkeys (essentially a controlled damped harmonic oscillator). The hands-on sessions teach the students the basics of programming using a visual language, and offer those who struggle with written programming languages an alternative way of programming.

Programming with Arduinos
Paul Cruickshank University of St Andrews

The Arduino (www.arduino.cc) is a low-cost open-source microcontroller platform and programming environment. Primarily employed by hobbyists, it can also be used in a range of teaching environments. The platform features analogue and digital inputs and digital outputs and may be easily interfaced with sensors and peripherals such as accelerometers, LEDs and actuators. Programs are written using a subset of C/C++ and platform-specific functions make it simple to address the inputs and outputs. Once a program is compiled and uploaded to the platform, it runs independently of the host computer.

The ease with which the Arduino can interface with and control external peripherals allows for a more interactive approach to teaching fundamental programming concepts than that afforded by traditional “command-line” based activities. We have developed, trialed and run an introductory programming lab (total length of 7.5 hours) for Scottish 2nd year physics students, most of whom have little or no previous programming experience, which culminates in students writing and using a program to automatically measure the IV characteristic of a diode. Evaluation capturing various aspects of the student experience found the lab to enhance student motivation and interest and the exercise met our objectives of introducing students to fundamental programming concepts such as loops and decision making. I will discuss practical aspects of implementing such a course, as well as the merits (and pitfalls) of the Arduino as a tool for introductory programming.

Creating usefulness when teaching programming
Rob Miles University of Hull

Great programmers are not just good at writing software. They produce solutions that are genuinely useful. In this talk we will explore the concept of usefulness as applied to programmers and consider how the challenge of providing a useful solution provides a good context for teaching programming fundamentals.

SEPnet Summer Placement Survey – what software do students use on placements and how well do physics courses prepare them for using industry software?

As part of the SEPNet 2014 Summer Placement scheme (2nd and 3rd year non- final year undergraduates) 54 students were interviewed and 43 employers from a range of industries including defence, energy, engineering and technology as well as research institutes. 10 large corporates, 23 SMEs and 10 research/education institutions participated. Approximately half of the projects comprised data analysis, modelling, programming. Other projects included testing products, website development, creating science demonstrations and research. Employers were asked how they rated students' software skills from 1-4, Students were asked: what software programs they had used previously, what they used on placement and how easy they found them to learn. The results of this study will be presented.

Participants

Steve	Barrett	University of Liverpool
Ed	Bennett	Department of Physics, Swansea University
Christopher	Booth	University of Sheffield
Ashley	Cadby	The University of Sheffield
Davide	Costanzo	University of Sheffield
Michael	Croucher	University of Sheffield
Paul	Cruickshank	University of St Andrews
Louise	Dash	University College London
Chris	Dewdney	University of Portsmouth
Janette	Dunn	University of Nottingham
Rachel	Edwards	University of Warwick
John	Fenner	University of Sheffield
Ross	Galloway	University of Edinburgh
Paul	Howes	University of Leicester
Katherine	Inskip	University of Sheffield
Mark	Jones	The Open University
Stuart	Littlefair	University of Sheffield
Myfanwy	Lloyd	University of Manchester
Andrew	Markwick	University of Manchester
Keith	McKenna	University of York
Rob	Miles	University of Hull
David	Mowbray	University of Sheffield
Andrew	Narracott	University of Sheffield
Adam	Petrus	STEM/BAE Systems
Julian	Pittard	University of Leeds
Matt	Probert	University of York
John	Richer	University of Cambridge
David	Sands	University of Hull
Niels	Walet	University of Manchester
Jim	Weston	University of Sheffield
Nicola	Wilkin	University of Birmingham
Anna	Wilson	University of Stirling
Sergei	Zharkov	Hull University