

Queen's University Belfast builds reliable, publishable Python capability for research.



Queen's University Belfast needed researchers and PhD students from multiple disciplines and institutions to work within shared codebases and produce reliable, maintainable, publishable code. Learning Tree designed and delivered a bespoke Python programme, led by consultant Sean Smith, that took a single cohort of 19 through two sequential phases — building core Python foundations before progressing to advanced software engineering practice. The result: consistently high evaluation scores and stronger, more consistent research code across the community.

From core foundations to advanced engineering practice

Research that spans multiple disciplines and institutions faces a challenge a single course cannot solve. When contributors to a shared codebase work to different standards, quality becomes inconsistent, code is harder to maintain, and the reliability of published outputs is put at risk. Queen's University Belfast recognized this.

As new researchers and PhD students joined the RIED (Re-Imagining Engineering Design) project, the University set a clear objective: establish a consistent coding standard so that every contribution, especially the code behind published findings, was well-written, reliable and able to stand on its own.

Sean was highly knowledgeable, engaging and enthusiastic, and I learned a lot from the course.

— Introduction to Python Programming participant

The programme delivered on that objective. A single cohort of 19 researchers completed both phases together, with evaluation scores consistently high — up to 3.93 out of 4 for course activities and instructor performance. Participants left able to apply testing, version control and clean-code practices directly to their research.



3.9 / 4
Overall activities
evaluation

3.7 / 4
Overall course
evaluation

3.8 / 4
Overall content
evaluation

3.9 / 4
Instr. Knowledge
of subject area

3.9 / 4
Overall Instructor
Evaluation

3.7 / 4
Instructor Real-
World Insights

2

Sequential phases — core Python,
then advanced practice

19

Researchers in one cohort, start to
finish

5

Days of hands-on virtual training
across the programme

The challenge: consistent, publishable code across a shared codebase

The RIED project brings together researchers and PhD students from different disciplines and institutions, all contributing to a common codebase. Earlier coding training had helped the initial project group, but as new people joined, individual work needed to meet a consistent standard, especially the code that accompanies published research.

Several specific challenges required a structured response

- Researchers needed to reach a professional standard together, confident in core Python before taking on advanced engineering practices.
- Researchers needed modern engineering practices; version control, automated testing and clean code, to make their outputs maintainable and publishable.
- The project needed a consistent coding standard across contributors, so that work from different disciplines and institutions fitted together reliably.
- Training needed to fit around demanding research schedules and relate directly to real-world tasks such as Computer-Aided Design (CAD), simulation, and data analysis.
- A single one-off course would not have been enough to build lasting capability. The cohort needed to build confidence in the fundamentals before progressing to advanced practice.
- Queen's University Belfast needed a tailored approach that built the cohort's core skills first, then advanced them into professional engineering practice.

The Learning Tree solution: a structured, two-phase programme

Learning Tree structured the programme in two sequential phases so the cohort could build core skills before advancing. Consultant Sean Smith led throughout, delivered online with pre-configured virtual machines and hands-on exercises, and adapted the pace to the group.



One cohort of 19 - the same group started together and finished together, progressing from core foundations to advanced practice.

Practical application ran through both phases. Around 60% of each session was taught, with the remainder spent on supervised, hands-on exercises in a virtual lab — so participants could apply concepts immediately to their own research code.

Core practices reinforced through the programme included:

- Clean coding in Python, including safe coding and tools such as PyLint and Ruff.
- Test automation with Test-Driven Development, PyUnit, PyTest and mocking.
- Version control and collaboration using Git and GitHub, from branching to pull requests.
- Applying these practices across services, REST APIs and containerised environments.



The results: strong scores and immediate research application

Measurable learning outcomes

Participants consistently praised the clarity of instruction, the logical progression of the material, and the value of practicing in the virtual lab. Even experienced coders reported gains. One noted the course **'filled a lot of basic knowledge gaps despite my use of Python for nearly 6 years.'**

The hands-on exercises reinforced the concepts effectively and helped build confidence in applying advanced Python techniques.

— Python Best Practices participant*

	Introduction to Python (2 days)	Python Best Practices (3 days)	Impact
Content	3.86	3.80	A strong foundation enabled advanced practice
Activities	3.93	3.86	Skills applied directly to publishable research code
Instructor	3.87	3.93	Kept advanced topics accessible and built trust
Overall	3.73	3.64	

The programme scored consistently well across every evaluation dimension. Both courses were evaluated by participants on a 0–4 scale.

From training to applied research capability

The defining outcome was the shift from learning to applied capability. Participants left able to structure and maintain their code, manage it with version control, and test it to a professional standard.

1. Capability	2. Confidence	3. Consistency
A structured progression from core Python fundamentals to advanced software engineering practice, taken by the whole cohort.	Instructor-led teaching and hands-on virtual labs helped researchers apply new skills directly to their own code.	Shared standards and a common vocabulary support reliable collaboration on the RIED project's shared codebase.

Several participants identified immediate uses in their own work, from module development and packaging to building reusable libraries for demanding computational research.

We're Built on Excellence, Impact & Innovation

For more than 50 years, Learning Tree has helped organizations build real-world capability through training that delivers measurable results. The difference is a consistent focus on application: instructor-led delivery from practitioners with genuine industry experience, programmes designed around each client's needs, and learning paths that build skills people can use immediately.

The evidence from this engagement is specific. Evaluation scores up to 3.93 out of 4. One cohort of 19 taken through two sequential phases. A research community better equipped to produce reliable, maintainable and publishable code.

Ready to build tailored technical capability for your team?
Contact a Solutions Manager today.



888-843-8733 • [Learningtree.com](https://www.learningtree.com)

*Participant quotes are drawn from anonymized Learning Tree end-of-course evaluations.