Programming, Bridging Course (1TD046)

- Comprised of 2 parts:
  - Script Programming (Python, 1TD328)
  - C programming (also 1TD046)
  - Possibility to take only the Python script programming part (1TD328).
  - Both parts share a site on studentportalen.

- Both parts are:
  - Taught as labs, with few seminars.
  - Programming Assignments (these will be reviewed in class).
  - No formal lectures, no traditional written exams.

- Teaching
  - We'll work through example problems in a supervised lab environment.
  - We'll learn about programming by writing, running and debugging Python and C code.
  - i.e. its self taught by *getting your hands dirty*.
  - Supported by coding demonstrations.
Script Programming (Python, 1TD328)

- **Assessment:**
  - 5 Credits.
  - 2 'Homework' Assignments, (pass/fail). Mandatory.
  - 1 Larger Assignment. Mandatory.
  - No written exam.

- **Content:**
  - Basic introduction to the Linux interactive shell, and programming with Python.
  - Use NumPy, MatPlotLib for analysis of datasets.
  - Relevant theory is introduced, but focus is on using programming to solve real problems in scientific computing -- rather than programming theory.

- **Instructor:**
  - Ben Blamey (Dr.)
  - ~10 years professional programming experience in private sector / academia. Now PostDoc in TBD.
  - Taught in English. (my Swedish is terrible...).

http://www.uu.se/en/admissions/master/selma/kurspian/?kpid=35881&type=1
Programming, Bridging Course (1TD046)

● What we’ll learn:
  ○ Compiled languages (C)
  ○ How to build and maintain a scientific project using both Python and C
  ○ More advanced use of the Linux Shell, version control, debugging, ...

● 10 credits (including 5 from the script programming part)

● Examination:
  ○ Everything from the Script Programming Part
  ○ + 2 additional assignments
  ○ + 1 final project

● Instructor:
  ○ Adrien Coulier
  ○ PhD student, ~12 years of personal programming experience, including programming competitions at university
  ○ adrien.coulier@it.uu.se

http://www.uu.se/en/admissions/master/selma/kursplan/?kpid=35880&type=1
Script Programming (Python, 1TD328)
QUIZ!
Syllabus / Schedule

See:
https://studentportalen.uu.se/portal/portal/uusp/admin-courses/admin-courses-course?entityId=157680
Teaching Approach

- Learning by doing.
- Read the docs yourself.
- Try it first - talk about it afterwards.
- Coding demonstrations.
- Pair Programming
Learning Resources

- Use a cheatsheet! (see studentportalen, or search for your own).
- The docs! [https://docs.python.org/3.7/index.html](https://docs.python.org/3.7/index.html)

My recommended books.

Writing code is the best way to learn! (IMO…)

[https://automatetheboringstuff.com/](https://automatetheboringstuff.com/)
## Lab Exercises & Assessment

<table>
<thead>
<tr>
<th></th>
<th>Help*?</th>
<th>Team work?</th>
<th>Where?</th>
<th>Marking:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lab Exercises</td>
<td>Yes</td>
<td>Encouraged!</td>
<td>In Lab</td>
<td>Not marked. Informal feedback in labs (time permitting).</td>
</tr>
<tr>
<td>Assignments 1 + 2</td>
<td>No</td>
<td>No - work individually</td>
<td>Outside Lab</td>
<td>Marked out of 5. 3 for a pass. Does not count towards final mark, (but you need to pass). Opportunity to retake if in case of fail. Reviewed in Seminars.</td>
</tr>
<tr>
<td>Project</td>
<td>No</td>
<td>No - work individually</td>
<td>Outside Lab</td>
<td>Marked out of 5. 3 for a pass. The overall mark for the course is the project grade (conditional on completing assignments 1 + 2). Opportunity to retake in case of fail, for max of 3. You may be asked to complete a follow-up oral assessment, based on your project work.</td>
</tr>
</tbody>
</table>

* Help with tasks themselves. Help is always available for clarifications and errors in the assignment.
Notes on Assignments

- Read the mark schemes and submission instructions carefully.
- Assignments will have optional questions -- only attempt these if you have completed all previous questions, and happy with the answers.
- *Because of the scheduling - some of the deadlines are quite tight!*
Help! I'm Stuck?  

Lab time is very limited...

1. Read the docs! Check the details -- is that upper bound inclusive or exclusive? Double check the order of the parameters when calling a function.
2. Try copying an example from the documentation, and modifying it to your problem.
3. Work iteratively in small chunks. Write a part of your script, convince yourself it is correct, make sure you understand why it works - then add a little more.
4. Make life easy for yourself - keep your code clean, with sensible variable names & comments.
5. Read the error message, try to understand it, does it mention a line number (your error may be on one of the nearby lines).
6. Re-read your code, line by line, check the syntax! (most errors relate to syntax).
7. `print()` is your friend! Print out the values of your variables at different stages in your script.
8. Ask your neighbour for help (only for the lab exercises!) Try to explain what your code does.
9. Be ready to explain what the problem is, and how you’ve tried to fix it!
10. Test your own understanding by trying to explain to your neighbour.
About Me - Ben Blamey

Moved to Sweden 2015. ben.blamey@it.uu.se

My programming background:

● Self taught!
● Commercial software developer since 2008 (mixed with time in academia).
● Automotive / IoT, social media, some finance.
● PhD in data mining / scientific computing (social media, NLP), mostly Java, Python.
● Lecturer (Assistant Professor), Cardiff University -- lab programming teaching.
● Large Commercial Projects (mostly closed-source):
  ○ Windows desktop applications, code generation. (WPF, C#, C++)
  ○ Web (mostly backend), LAMP (Linux, Apache, MySQL, PHP).
  ○ Mobile Apps - both Android/Java and iOS/Swift.
  ○ Server-Side (NodeJS, JavaScript)
● Currently PostDoc here in TBD, working on HASTE Project:
  https://github.com/HASTE-project (Python)
FAQs

Which parts are mandatory (for students with conflicting schedules)?

- You are encouraged to attend all sessions. Lab sessions are an opportunity for feedback on your lab work!
- You are **strongly encouraged** to attend seminars (as we will review the previous work, and discuss any difficulties)
- **Because of the scheduling - some of the deadlines are quite tight!**

What will be posted on studentportalen?

- Worksheets, assignments, all other documentation.

Can I email you?
Yes, but please use the lab sessions to ask questions where possible rather than emailing me -- its very hard to discuss code over email!

Can I use my own laptop?
Yes, but please ensure you have a working **Python 3** environment and IDE setup before the lab session. Otherwise, use the lab machines.