Day 2

Morning session: 9:00 - lunch

- A closer look: Wednesday and Friday assignments (small groups)
- Tools (presentation/demonstration)

Lunch (13.00 – 14.00)

Afternoon session: 14:00 – 16:00

- TeachBooks Introduction and Demonstration
- Parallel sessions:
  - TeachBooks continued (more features; make your own book)
  - Git
  - Assessment

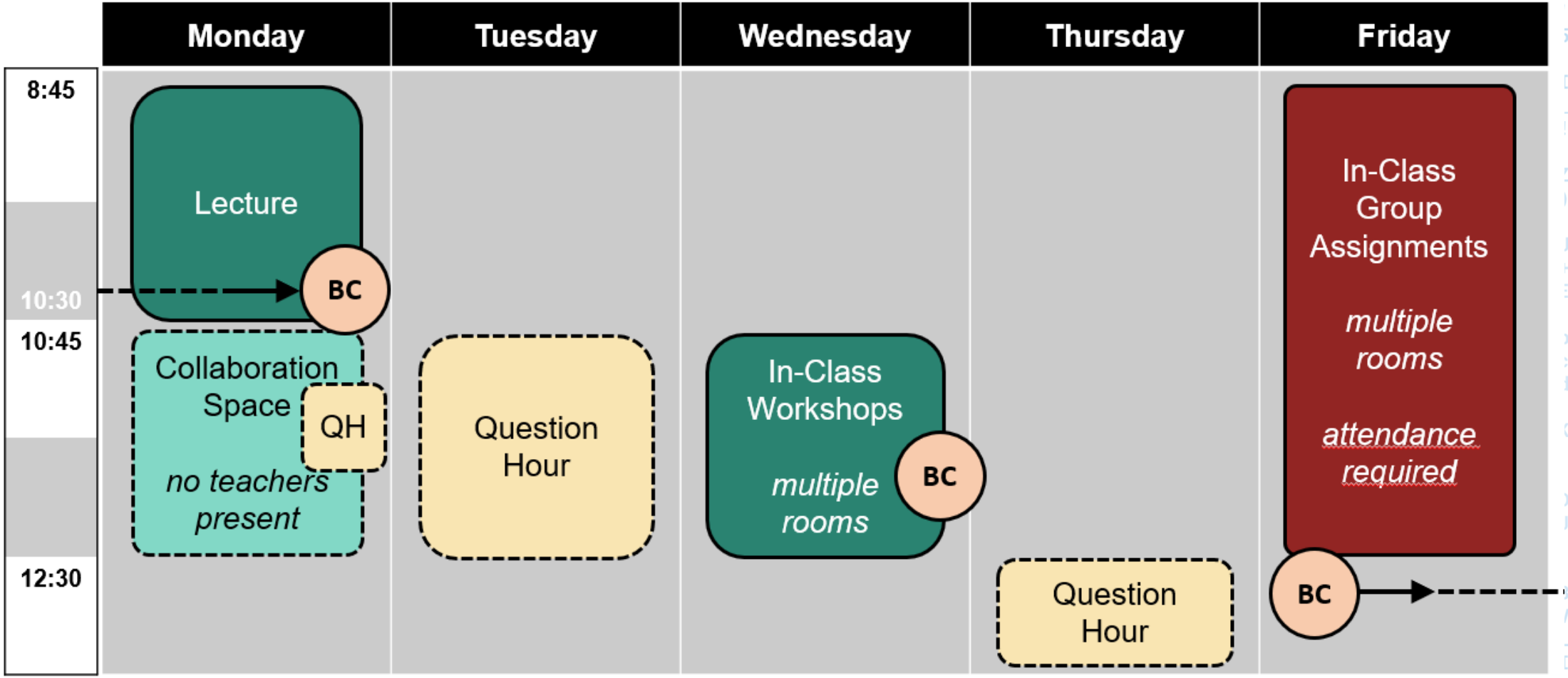
Closing: 16:30-17:00 (discussion / activity depending on time)

TU Delft

MADE

Weekly Schedule

Note: this is Year 3 (current year). Slightly different from Year 2 shown yesterday.



Programming Assignment: any time during the week, but... **Finish before Friday!**

BC = BuddyCheck: opens Fri (closes Mon); review results Wed with group

Question Hours (optional): Mon 11.00-12.00, Tue 10:45-12:30, Thu 12:30-13:30

Workshop Materials

- Demo book contained 2 weeks work of material
- Each week: programming assignment (PA), workshop (WS), group assignment (GA)
  - PA is graded individually pass/no pass
  - WS is in-class, not turned in
  - GA is in groups, turned in for a grade (parallel session later on assessment)

TU Delft

MDE

In-class session: workshop

- We run the in-class sessions with 4-5 rooms, and (ideally) 2 teachers per room (up to ~100 students)
- Activities that are not turned in for a grade are not deployed with GitHub
- Using a pptx slide in class is very useful
  (shared file; changes are updated everywhere)
- The next slide is what we used last week (with a blank link)

Note: we use GitHub Classroom for graded assignments; ungraded assignments can be downloaded from a website (we try to teach good practices for file storage so they can save their work; we also put “read only” versions of everything online as HTML to make it easy to review)

Instructions for Today

- Files are in: <website where files are located or link to GitHub Classroom>
- You don’t need to hand this in! Solutions will be made available this afternoon
- Remember the ANS review comment deadline is Friday 22nd!
  We will start reviewing your questions in ANS soon!
- PA status updated in BS weekly, but still can submit till end of Q2 (Monday, 2.9)

$$\left[\int \mathbf{B}^T E A \mathbf{B} + \mathbf{N}^T k \mathbf{N}\, dx\right] \mathbf{u} = \int \mathbf{N}^T q\, dx + \mathbf{N}^T F \Big|_{x=L}$$

should be:

$$\left[\int \left(\mathbf{B}^T E A \mathbf{B} + \mathbf{N}^T k \mathbf{N}\right) dx\right] \mathbf{u} = \int \mathbf{N}^T f\, dx + \mathbf{N}^T F \Big|_{x=L}$$

Instructions for IITM Workshop

- Files are in: iitm-mude.github.io/2024-workshop-files/students/WS1/
  - Read only → WS_1_5_dont_integr_hate.html
  - To use notebook, download the zip file

TU Delft

MUDE

Discussion

- General reaction: how was the WS and GA? Clarity, difficulty, length?
- Could you recognize how the PA topics were included?
- Challenges/insights in the use of Python
- Could you envision yourself creating such an assignment?

Some comments from our side:

- Use of list comprehensions was not ideal!
- We like the way the notebook can focus directly on the method
- There is a lot of material and preparation needed outside the notebook
- Technical experts can contribute easily (given team support)

TU Delft

MUDE

Discussion points

See summary page on workshop website for recap of key discussion points

Working with notebooks with students

- The WS and GA assignments in-class on their own computer, with an IDE → the book is for introducing and practicing with fundamental concepts
- File management can be complicated
  - Maintaining student version and version with solutions
  - Teachers need to be able to read the solution in class
  - Need a convention for creating the student version and correcting it reliably

- Student and Solution versions
  - Make all files as if they are the solution
  - At the very end, create the student version
  - Create a convention to easily see student version in the solution
  - The solution is also useful for teachers (e.g., revising material next year)

In-class session: group assignment

- GA1 on Files page: tudelft-mude.github.io/workshop-iitm-files/students/GA1
- Just read the HTML files → 10 min, then discussion

- Our “typical” set of files is README.md - Analysis.ipynb - Report.md
- Pretend you are a student for the reading part (no time for code now):
  - Start with README
  - Read Analysis
  - Read Report
  - Look through solution files, feedback file (given after grading)

TU Delft

MUDE

Follow-Up Assignment

- Next week we introduced debugging and testing
  → content in teachbooks.io/learn-programming/
    - assert statements as a way to check that code is working properly
- Used code from Friday as a case study

  → turns out many students did the assignment wrong without realizing it!
- Excellent example of code being useful for teaching theory
- This resulted in the “explanation” added to solution (see IITM Files page)

NOTE: the use of “real data” and non-uniform step sizes in GA to study numerical derivatives did not work so well. We will adapt it next year.