

BU-EC444

Fall 2024
Prof. Little

Course Expectations

Course sites

Course E-Book: <https://github.com/BU-EC444/01-EBook>

Course repositories and reporting: <https://github.com/BU-EC444>

Course Grading / Blackboard: <http://learn.bu.edu/>

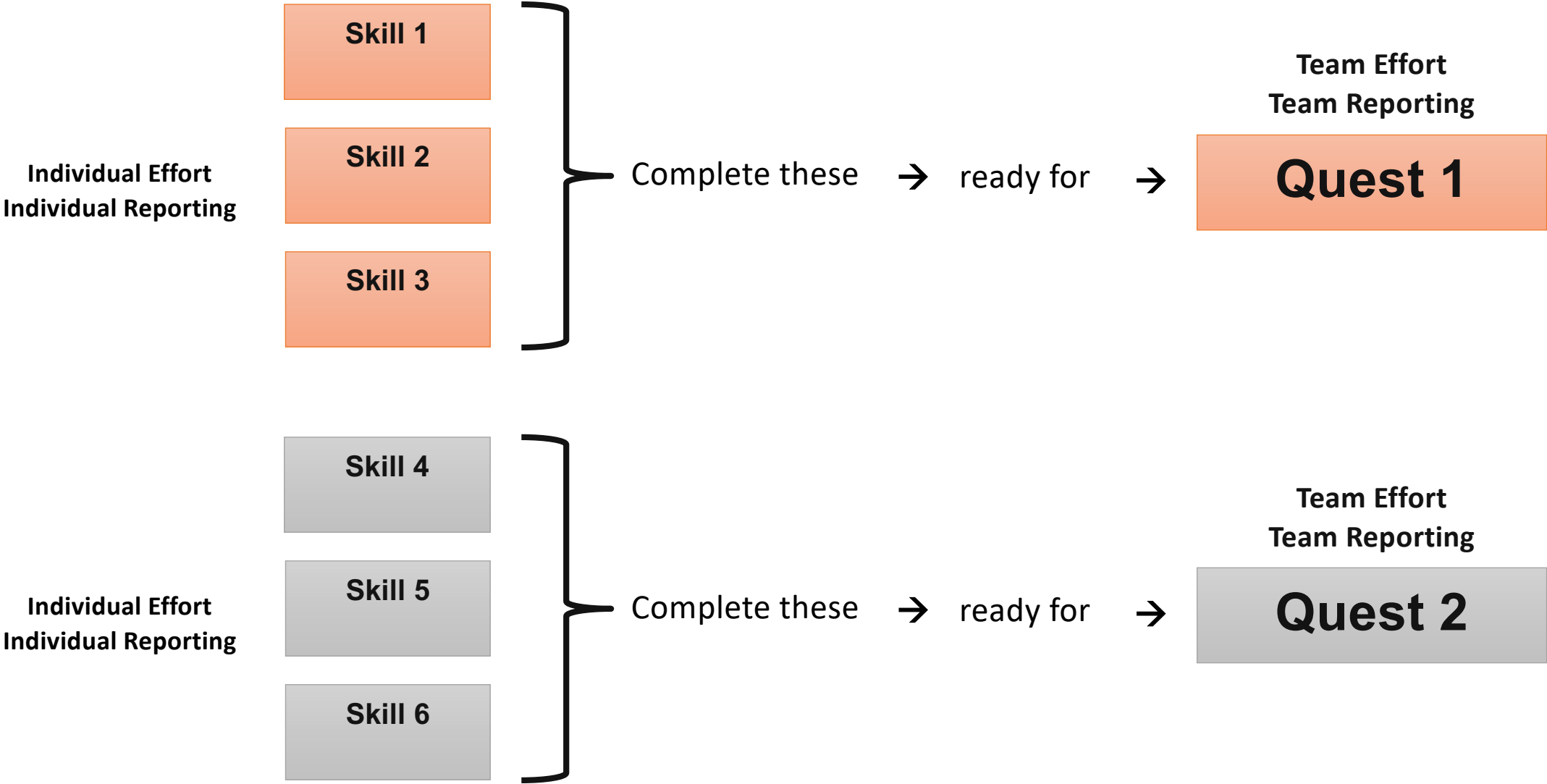
Course Discussion: <https://piazza.com/bu/fall2024/ec444>

Prerequisites

- Basic computer organization
- Programming in C
- Design of a software solution for a system
- Working with basic electronics

Critical: we will program in C

Workflow – Quest clusters



“No Lectures”

- Class time is for working on quests
- Skills parceled-out as we go with brief introductions and tips to material
- Remainder of time will be rotating visits to your teams in the studio lab

It is essential that you collaborate with your team

Class Flow -- approximately

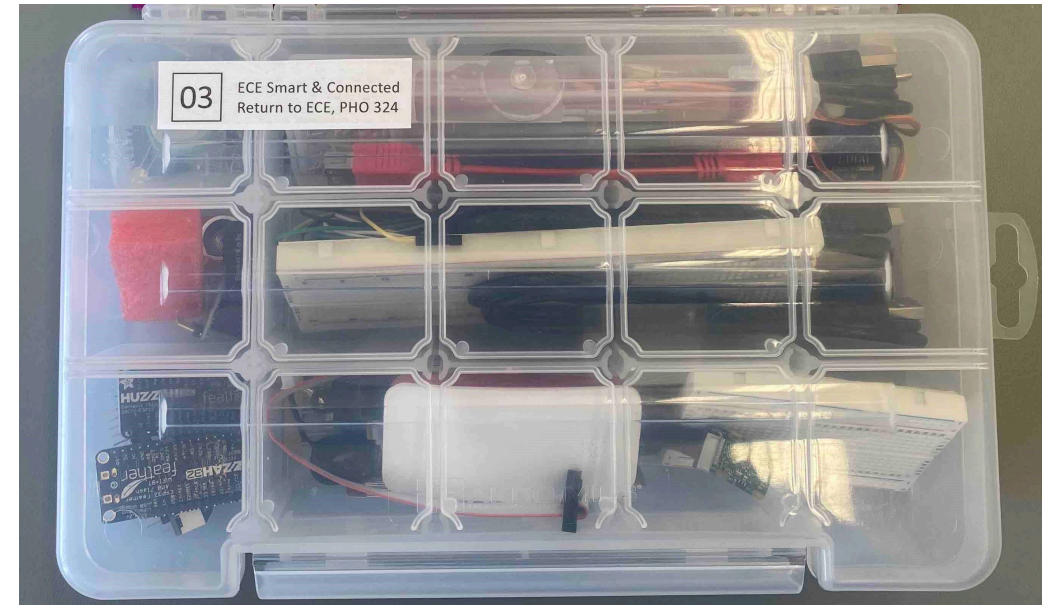
1. 10 min introducing new topics or overviewing the tasks for each class
2. Students will break into teams to complete tasks (skills and quests)
3. Instructor and the TA will rotate through meeting with teams for the duration of the session; depending on the session we may service groups based on demand requests
4. I'll mix in some live interaction with respect to the current assignment to incentivize in-class task completion
5. **You will be expected to participate in every session unless explicitly excused (no sleeping-in). Your team depends on this participation**

Kits

- >\$500 in parts, hand assembled
- Group kit
- Individual kit

Kits must be returned

- At the end of the course, or
- Immediately if you drop the course



Scoring

1. 6 quests, 100 points each → 600 points total
 - 75% objective assessment (it works or not)
 - 25% qualitative (design, reporting etc.)
2. 6 skill clusters of ~6 skills, 10 points/skill → approx. 360 points
 - Show evidence of successful completion: 10 pts
 - Modest attempt: 5 pts
 - Weak or no attempt: 0 pts

Expectation is that effort and completion of assignments will lead to higher grades

Late assignments and re-dos

- Quests and skills in quest clusters are due at 11:59PM on dates assigned.
- Your team can get **one exception** for up to 5 days on submitting a quest
- Late assignments will be discounted by 10%
- Must notify instructors of exception, via Piazza
- Individual assignments: you can **resubmit** skill assignments up until the end of Thanksgiving break (12/1/24). You must call out changes and notify instructors via Piazza if you want a regrade

Timing

1. 14 weeks, 3 studio blocks per week: ~40 blocks
2. 6 quests (plus installs), or approximately 1 quest every 6 blocks

The course is quest based – this is where the points are, but you complete the individual assignments (before) the quests

Reporting

- Skills – skills template using markdown
- Quests – quest template using markdown
- Use of images and sketches
- Use of video
 - For quests: up to 120s for overview plus up to 120s of demo
 - **Landscape mode only**
 - Best videos will be shared with class with permission
 - Videos stored on Google Drive (not on GitHub)
- Reports should be developed locally, on your laptop(s) and pushed to GitHub
 - Don't push build files please – exclude these in .gitignore

Other details

Collaboration

- In teams – yes
- **Individual assignments must be done individually** – everyone needs to show evidence of completing the skills
- If you use code found on the web: must be attributed with URL and you will need to be prepared to defend it

Generative AIs

- Can use generative AIs **with conditions**

Our experience with ChatGPT so far

- Works with ESP32 code, not trustworthy
- Works better on algorithms or more commonly used code
- Clearly an emerging tool to support design, development, debugging
- Valuable to include ChatGPT as a tool in the toolbox, but with some rules

There is risk that you will not learn fundamentals if you rely on AI

- Using AI may reduce your competency for completing a technical interview
- It is important to understand a problem solution
 - To cast a prompt
 - To evaluate the result
- When reusing code, students tend to repeat existing code rather than build modular code. This leads to other problems including resource conflicts and performance degradation.

Generative AIs

Conditions for use

1. You must indicate in your reports when you have used an AI to produce a block of code
2. The **block should be cited in code comments** when it has been produced by the AI
3. You are responsible for understanding the results that you adopt; the expectation is that you can defend the solution and justify its correctness

Other requests

1. [Skills] Start with example code and try to understand how it works rather than going straight to AI
2. [Quests] Use a design process model and then synthesize your problem and then your solution. This is best on pencil and paper or whiteboard with participation by your team. Divide and conquer
3. [Quests] Use design patterns -- these are intended to be translatable into different contexts and language models
4. [Quests] Identify test cases for your individual pieces and your overall solution. Don't forget the edge cases.
5. [Quests] Beware of clumping repetitive code into solutions. This will be assessed as poor quality of your solution when detected.

Other details

Class schedule

- On e-book <https://github.com/BU-EC444>
- Includes quest due dates, cluster skills due dates, demo dates
- **First deliverable due on Friday**

Instructors / TAs

- Communicate over Piazza
- Prof. Little: OH Tu/Th
3:30—4:30PM
- TAs:
 - Benjamin Gilbert
 - Noah Robitshek



Thomas Little

Dive into material at next class

- Need GitHub login IDs (post to Piazza)
- Start with Quest 0

- Bring your laptop with dongle to connect to USB 2.0 (USB-C to USB-A)
- **Macs: please install Xcode in advance** (slow)