MY470_wk1_admin

September 30, 2024

0.0.1 MY470 Computer Programming

1 Welcome to Computer Programming

1.0.1 Week 1 Admin

1.1 Overview

- Course content
- Prerequisites and materials
- Course meetings
- Assesment and collaboration
- Weekly schedule

1.2 Why Do Social Scientists Need Computer Programming?

1.3 Why Do Social Scientists Need Computer Programming?

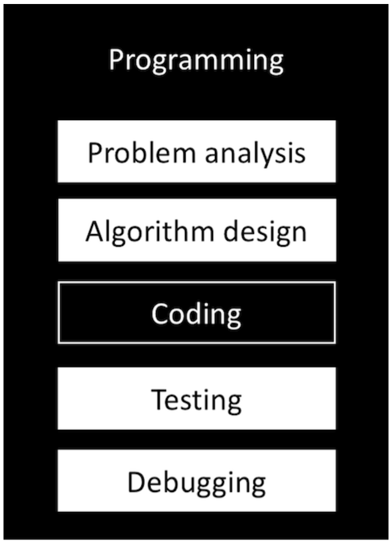
- Collect data

  - Crawling websites and using APIs
  - Online surveys and experiments
  - Computational models and simulations
- Manage, analyze, and visualize data
  - Large data
  - Non-rectangular data (e.g. networks, text)

- Be autonomous and work independently
- Learn from and collaborate with engineers and scientists

- Generate and share reproducible workflows

1.4 Coding vs. Programming



1.5 Course Content

- Introduce the fundamentals of computer programming
- Cover the foundations of computer languages, object-oriented programming, and algorithms
- Learn how to design, write, and debug computer programs and how to evaluate algorithms
- Practice on applications from computational social science and social data science

1.6 Instructors

- Milena Tsvetkova, `m.tsvetkova@lse.ac.uk`
- Daniel De Kadt, `d.n.de-kadt@lse.ac.uk`
- Yuanmo He, `y.he54@lse.ac.uk`
- Sia Shahrizad (GTA)

Course administrator * Serena Ngai, `methodology.admin@lse.ac.uk`

1.7 Prerequisites and Software

- Introductory course — no prerequisites

- Laptop for in-person classes
- Software
  - **Python** (Anaconda distribution) to learn basic concepts in computer science
  - **R** to experience another common programming language
  - **Visual Studio Code** and **RStudio** to write code
  - **GitHub** to share course documents and assignments

1.8 Course Materials

1.8.1 Lecture/seminar materials and sample answers to assignments

- https://github.com/lse-my470/lectures
- https://github.com/lse-my470/example-answers

1.8.2 Books

- Guttag, John V. *Introduction to Computation and Programming Using Python: With Application to Understanding Data.* Mit Press, 2016.
- Miller, Bradley N. and David L. Ranum. *Problem Solving with Algorithms and Data Structures Using Python.* Available at https://runestone.academy/ns/books/published/pythonds/index.html.
- Grolemund, Garrett and Hadley Wickham. *R for Data Science.* O'Reilly, 2016. Available at http://r4ds.had.co.nz.

1.8.3 Additional resources

- MY470 additional resources

1.9 Course Procedure

1. Read required readings
2. Attend lecture
3. Attend class
4. Complete and submit problem set on GitHub

- (Schedule office hours on StudentHub to chat about programming and research, ask questions about the course content, or get clarification about feedback)
- (Post and answer clarifying questions about assignments on Moodle)
- (E-mail Milena if you have an urgent problem, e.g. cannot access GitHub to submit assignment)
- (E-mail methodology.admin@lse.ac.uk if you require a deadline extension)
- (Use additional resources to practice more)

1.10 Course Meetings

- Ten 2-hour lectures
  - Mondays 13:00–15:00 in CKK.2.04
- Ten 1.5-hour seminars
  - Tuesdays 10:00–11:30 in CKK.1.09
  - Tuesdays 11:30–13:00 in CKK.1.09

    - Tuesdays 15:00–16:30 in CBG.2.02
    - Tuesdays 16:30–18:00 in CBG.2.02
- No lecture/seminar in Week 6
- Office hours
    - Book on StudentHub

1.11 Assessment

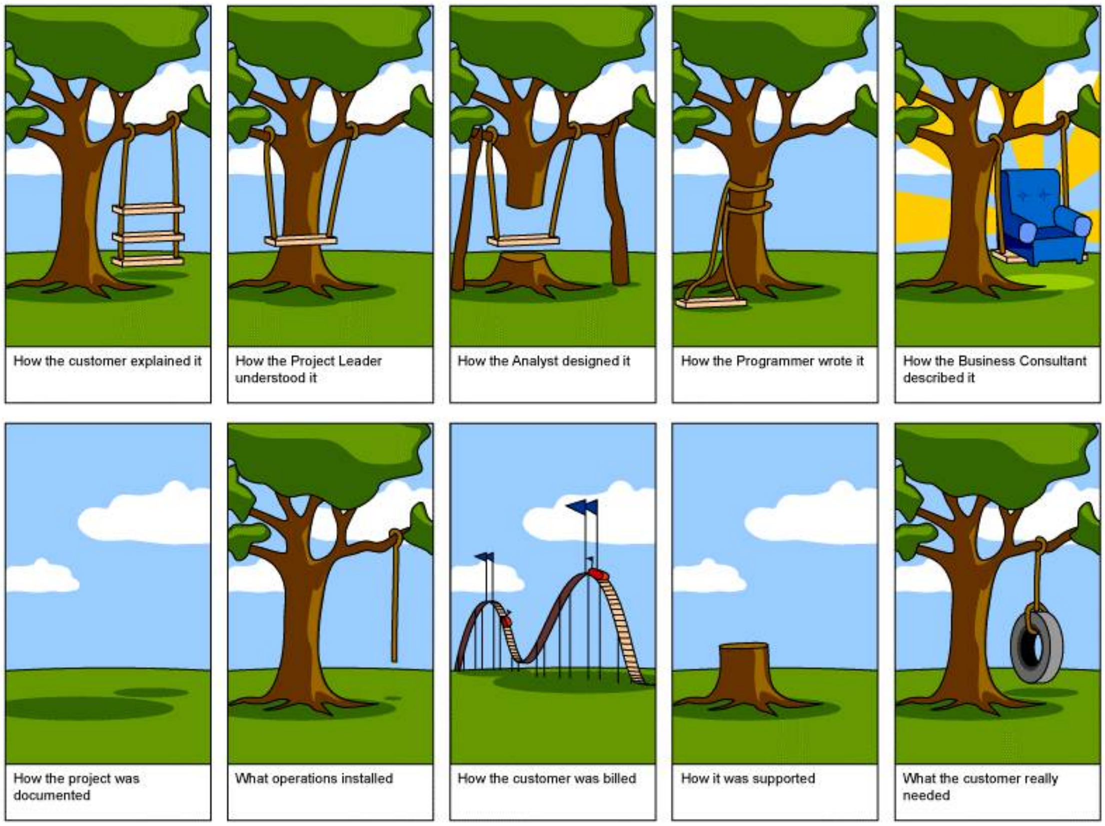
- In-class assessment (50%)
    - 5 problem sets
        - PS1-4 due at 12:00 on Mondays of weeks 4, 5, 6, and 8
        - PS5 to complete in class on the Tuesday of week 10
    - We will try to give you grades and comments by Friday

- Take-home exam (50%)
    - Substantive Python project requiring you to demonstrate concepts and skills learned from the course
    - For students taking MY570: need to come up with own project, talk to Milena
    - Due at 12:00 noon on Monday, January 20, 2025

1.12 Assessment Criteria

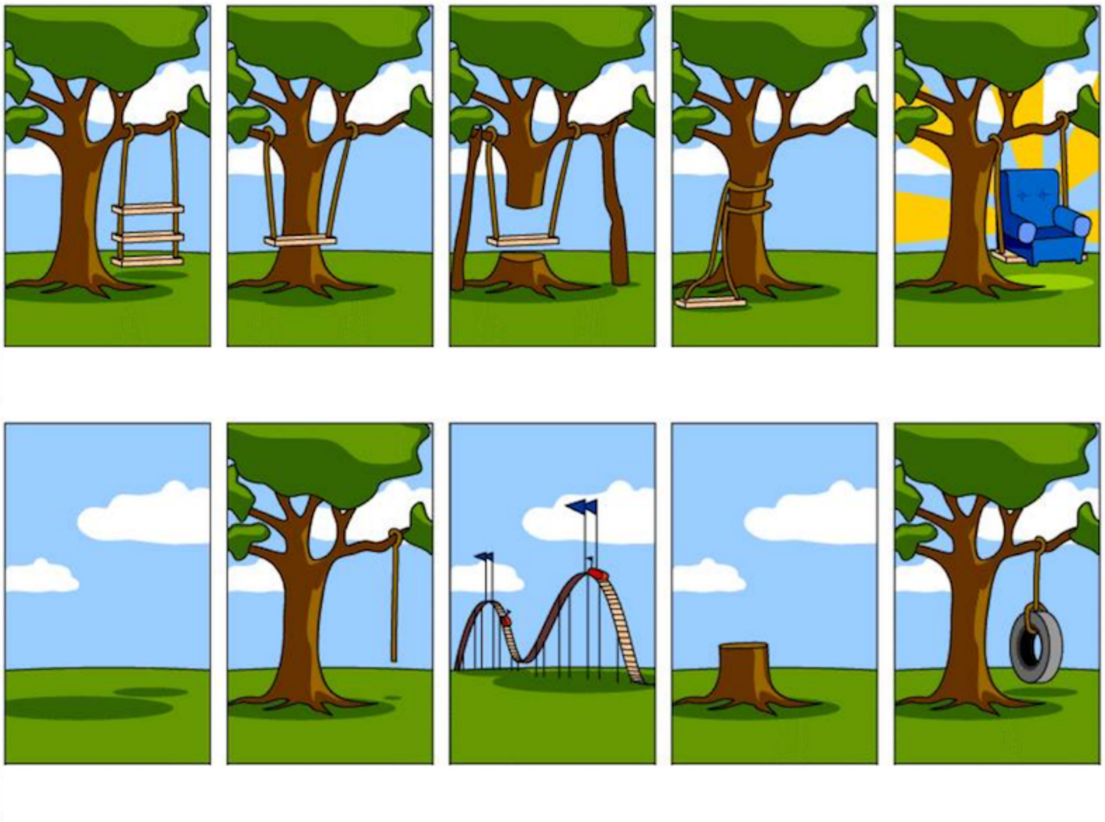
- The code runs and does what it is expected to
- The code is written using the concepts, paradigms, and best practices covered in the course
    - Legibility
    - Modularity
    - Optimization

1.13 A Classic Software Development Joke

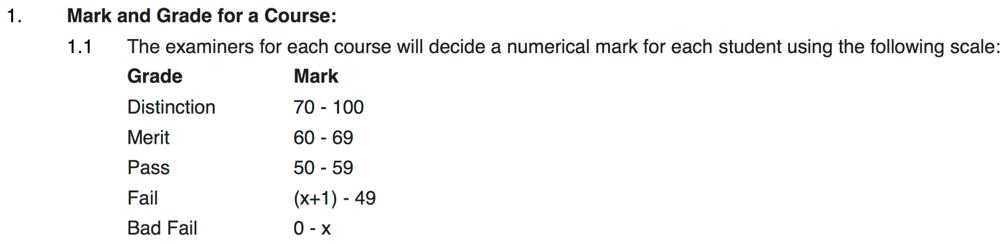

How the customer explained it
How the Project Leader understood it
How the Analyst designed it
How the Programmer wrote it
How the Business Consultant described it
How the project was documented
What operations installed
How the customer was billed
How it was supported
What the customer really needed

Source: Reddit

1.14 Assignment to Design a Swing: Submissions



1.15 Marks at LSE



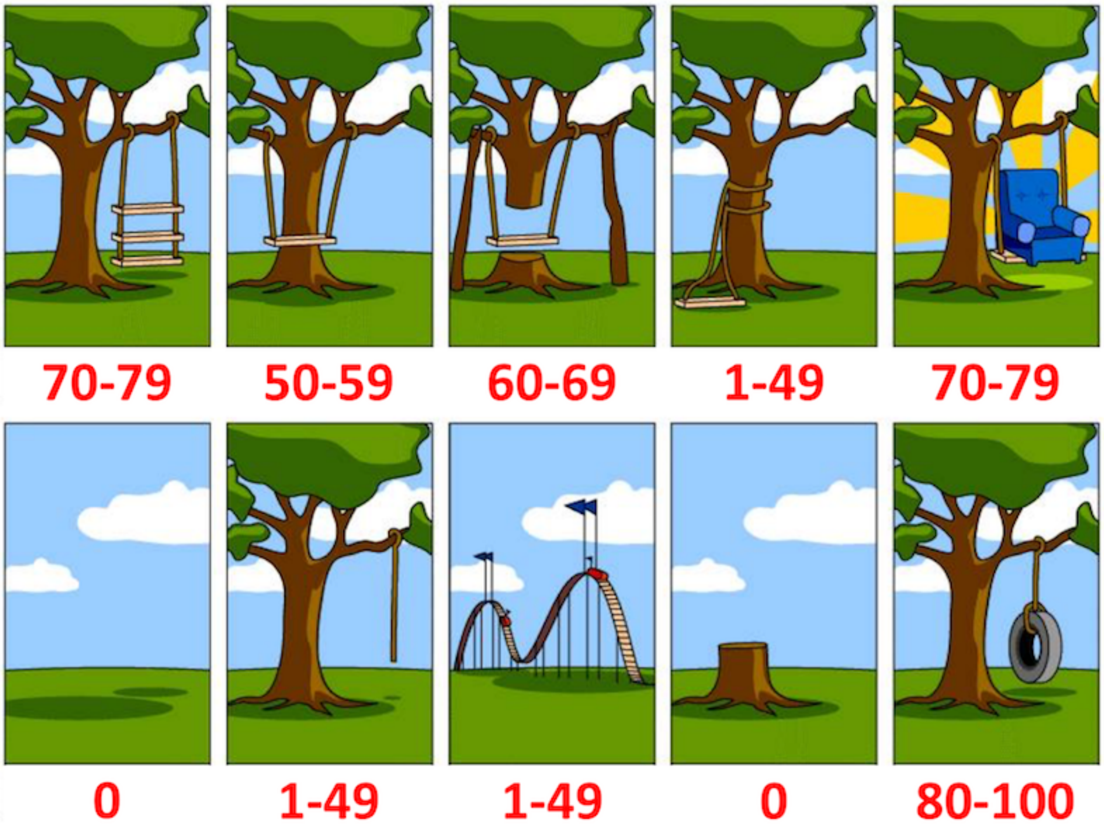
1. **Mark and Grade for a Course:**

1.1 The examiners for each course will decide a numerical mark for each student using the following scale:

Grade	**Mark**
Distinction	70 - 100
Merit	60 - 69
Pass	50 - 59
Fail	(x+1) - 49
Bad Fail	0 - x

https://info.lse.ac.uk/Staff/Divisions/Academic-Registrars-Division/Teaching-Quality-Assurance-and-Review-Office/Assets/Documents/Calendar/SchemeTaughtMasters.pdf

1.16 Assignment to Design a Swing: Marks



1.17 Collaboration and Plagiarism Policy

- Assignments are individual unless we instruct you otherwise
- For individual assignments:
  - You are not allowed to discuss solutions with peers, friends, family, or roommates
  - You are not allowed to show your code or view others' code (including solutions shared by previous students)
  - You are not allowed to ask questions about the assignments on Q&A sites such as Stack Overflow
  - You are not allowed to use AI tools such as ChatGPT and Copilot unless we instruct you otherwise
- You can search for general advice online (e.g. on Stack Overflow) but always give credit in comments if you borrow code

- You can use the forum "Clarifying Questions about Assignments" on **Moodle** to ask questions about the assignment instructions (no code allowed, however)

1.18 Anonymity Regarding Marking

- You will use your GitHub username to submit assignments
- GitHub account and activity are checked by employers so good to have an active account
- We use detailed marking criteria to evaluate your work and aim to be objective

- However, if you are worried about anonymity, choose/change your username accordingly
- See here for issues associated with changing GitHub username

1.19 Course Outline

Week	Language	Topic
1	–	What is Computation?
2	Python	Data Types
3	Python	Control Flow
4	Python	Functions
5	Python	Classes
6	–	–
7	Python	Testing and Debugging
8	R	Other Programming Languages
9	Python, R	Algorithms and Order of Growth
10	Python, R	Searching and Sorting Algorithms
11	Python, R	Tree and Graph Algorithms