Programming Methodologies: Python - Practicum 01

Topics Covered: Computational Thinking, Problem Definition, Abstraction, Algorithms, Flowcharts.

Learning Objectives:

- Apply computational thinking to solve complex problems.
- Break down complex problems into smaller parts.
- Create algorithms and flowcharts.

Deliverables:

- Submit a single PDF file named as `CET1011_P01_<Your_Name>.pdf` (e.g. `CET1011_P01_John_Doe.pdf`)

Overview

Computational thinking is a fundamental problem-solving approach that has become increasingly crucial in our technology-driven world. It is not just for computer scientists and programmers; rather, it is a mindset and skill set that empowers individuals across various disciplines to tackle complex challenges, analyze data, and make informed decisions.

In this practicum, we will apply computational thinking to frame problems in a way that allows us to express their solutions through computational steps and algorithms.

The Problem

You manage a research lab with multiple secure rooms. Scientists book rooms for 1-3 days and need exclusive access during their booking period.

During their booked period, no other scientists are allowed to enter except for the scientist who booked the room. The current access system relies on physical keys, which you manage, but it has its issues.

Current Issues with Physical Keys:

- Scientists lose keys or forget to return them
- Risk of unauthorized access from copied/stolen keys
- Expensive to replace locks frequently if keys stolen

The Supposed Solution 🤔

You've considered using electronic door locks and access cards with unique PIN-like codes generated for each scientist every time they book the room. The initial design of the electronic card system works as follows:

- Each card has a secret code, and each door lock will have a corresponding or matching code set inside. The code in the door lock has to be manually preset on the lock itself every time a new booking is made.
- If the code on the card matches the code set inside the door, then access to the lab is granted.
- Each scientist is given a new card with a new code set by the computer when they collect the access card from you.

However, this design has flaws - preset each door's code manually is impractical. Imagine if there are 50 rooms or more and you have to do this frequently.

To overcome this flaw, you extend the current idea that involves linking all the doors to a computer network, which, in theory, could be a functional solution. It would involve sending a message to each door whenever a new code is assigned to a key, informing it of the updated access code.

However, this approach proves to be cost-prohibitive, as it requires transforming each door into a significantly more complex computing system, necessitating network connections. Moreover, there will be security concern about safeguarding the network from potential hacking attempts.

Your Challenge 😈

Your task is to devise a more efficient solution from your existing idea of using electronic cards and door locks that complies with the following constraints/requirements:

✅ Does not require network connectivity for the doors.

✅ Allows lab staff to manage access control without the need to physically visit each door.

✅ Prevent anyone holding cards with invalid or old booking access, from entering.

✅ Door locks can read cards, perform logic/algorithm you design and update itself only.

✅ No additional hardware (beyond cards and locks) is used.

Assumptions:

1️⃣ Rooms are fully booked each day

2️⃣ No overlap in bookings

3️⃣ Scientist enter and leave their booked room punctually

Present your solution in a short paragraph (3-5 sentences) and also provide your solution in algorithmic steps (**pseudocode**). For example:

```
Sample solution:
The solution is .....
....
....

Pseudocode:
1. Start
2. Read `code`,...
3. If `code`==XXX ...
4. ...
...
...
X. End
```

Finally, create a **flowchart** that corresponds to your solution.

🤓 **Hints for Thinking**:

- What must each card and door lock store inside?
- How does the stored information in card and lock interact with each other to grant or deny access?
- How to design the logic for door lock to check codes or update itself?