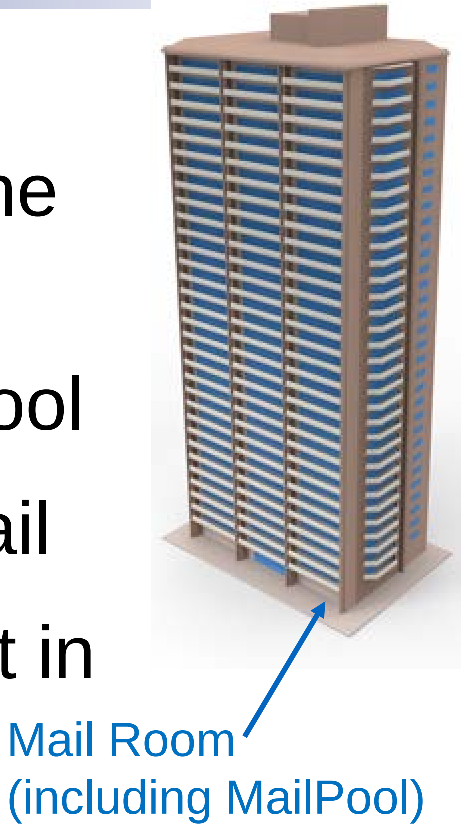
Project Part A: Mailbot Blues

- Mail Items arrive at arbitrary times at the Mail Room
- On arrival they are added to the MailPool
- Robot with a Storage Tube delivers mail
- Arrives at the Mail Room; items are put in its storage tube
- Starts delivery only when it says its ready
- *Deli*vers the items in Storage Tube order



Other Factors

- Priority – two levels 10 and 100
- Weight of mail varies
- Two robots provided
  - Each carries up to 4 items
  - New robots handles any weight
  - Old robot handles only light items
- You will work to defined interfaces
  - Organise mail in mailroom
  - Robot behaviour - limited aspects

Strategy Pattern

Problem

How to design for varying, but related, algorithms or policies? How to design for the ability to change these algorithms or policies?

Solution: (advice)

Define each algorithm/policy/strategy in a separate class, with a common interface.

MailBot Blues: Advice

- Start early
- Read the instructions carefully
- Run the package
- Read the relevant parts of the code (not much)
- Get a simple solution working
  - Test it
- Get a better solution working
  - Test it
  - Submit it