MARLON MOREIRA

Terminal App

Problem = Planning my day

Requires brain power

Takes time

Solution = My App

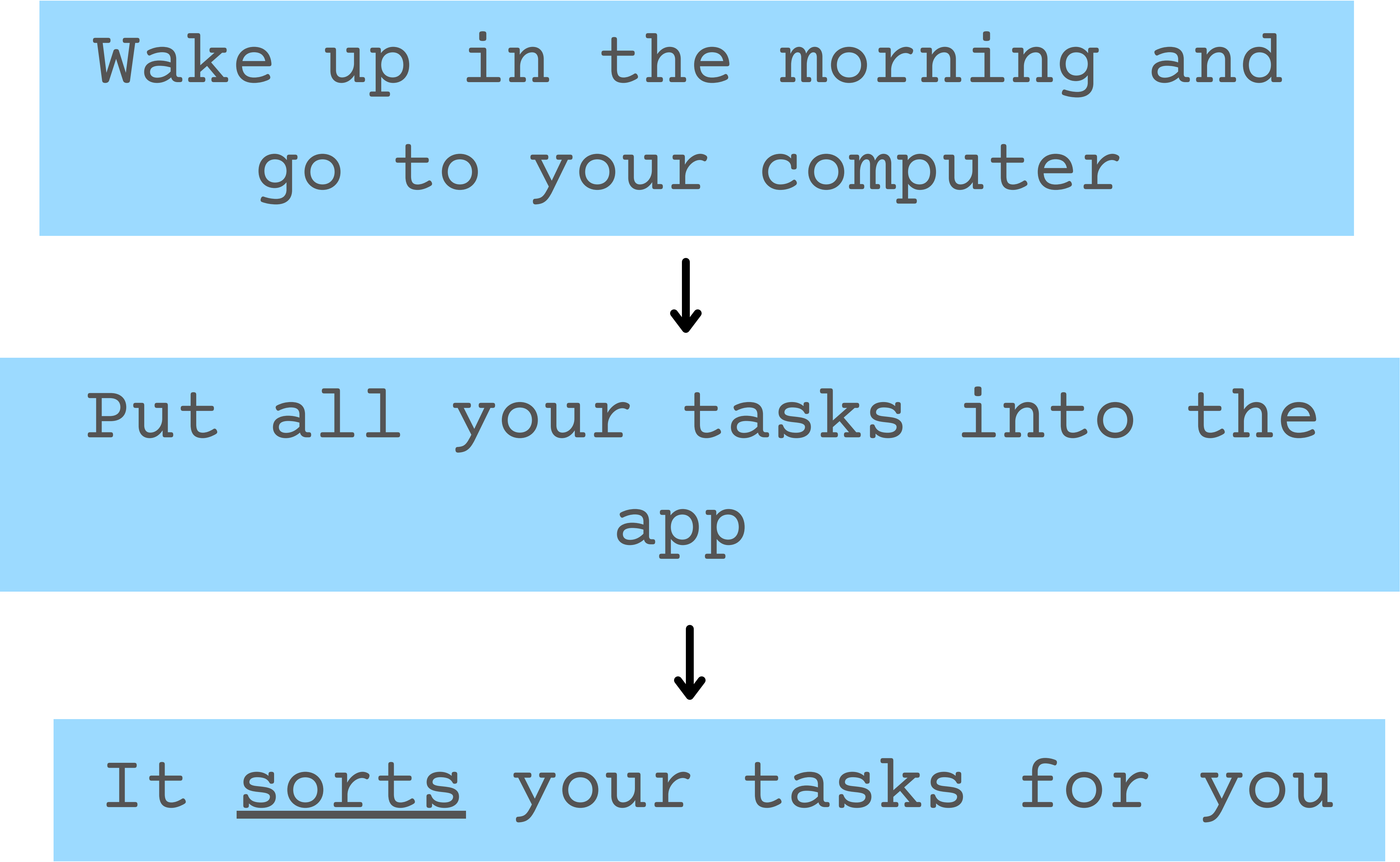
Plans my day for me

No brain power required

Saves time

```
                       _---~~(~~-_.
                     _{        )   )
                   ,   ) -~~- ( ,-' )_
                  (  `-,_..`., )-- '_,)
                 ( ` _)  (  -~( -_ `,  }
                 (_-  _  ~_-~~~~`,  ,' )
                   `~ -^(    __;-,((()))
                         ~~~~ {_ -_(())
                                `\  }
                                  { }
 _                   _             _
| |                 (_)           | |
| |__   _ __   __ _  _  _ __    __| | _   _  _ __ ___   _ __
| '_ \ | '__| / _` || || '_ \  / _` || | | || '_ ` _ \ | '_ \
| |_) || |   | (_| || || | | || (_| || |_| || | | | | || |_) | _
|_.__/ |_|    \__,_||_||_| |_| \__,_| \__,_||_| |_| |_|| .__/ (_)
                                                       | |
                                                       |_|
```

How does it work?



Wake up in the morning and go to your computer

↓

Put all your tasks into the app

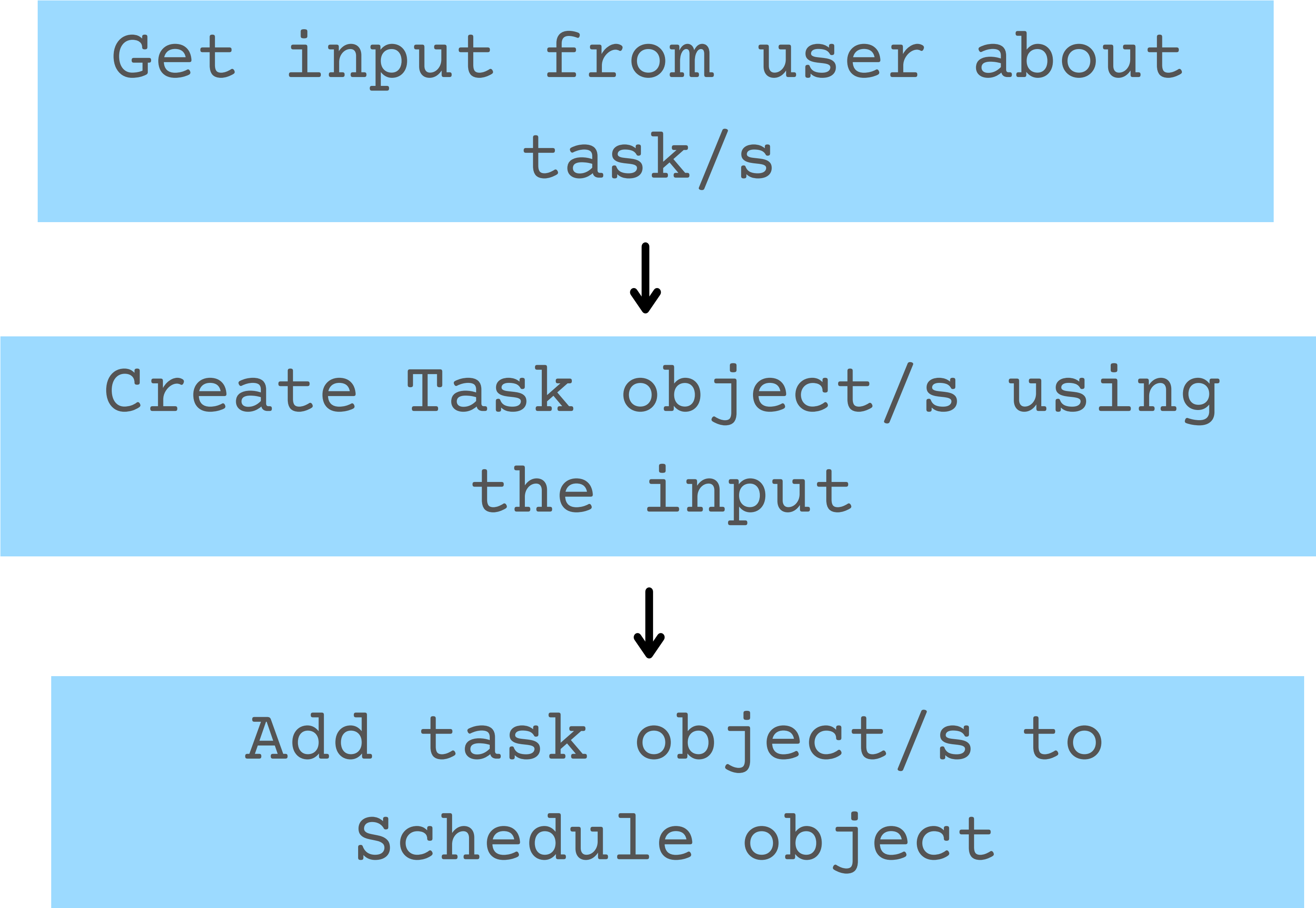
↓

It sorts your tasks for you

Feature 1: Add a task

(Demo)

Feature 1: Under the hood



Get input from user about task/s

Create Task object/s using the input

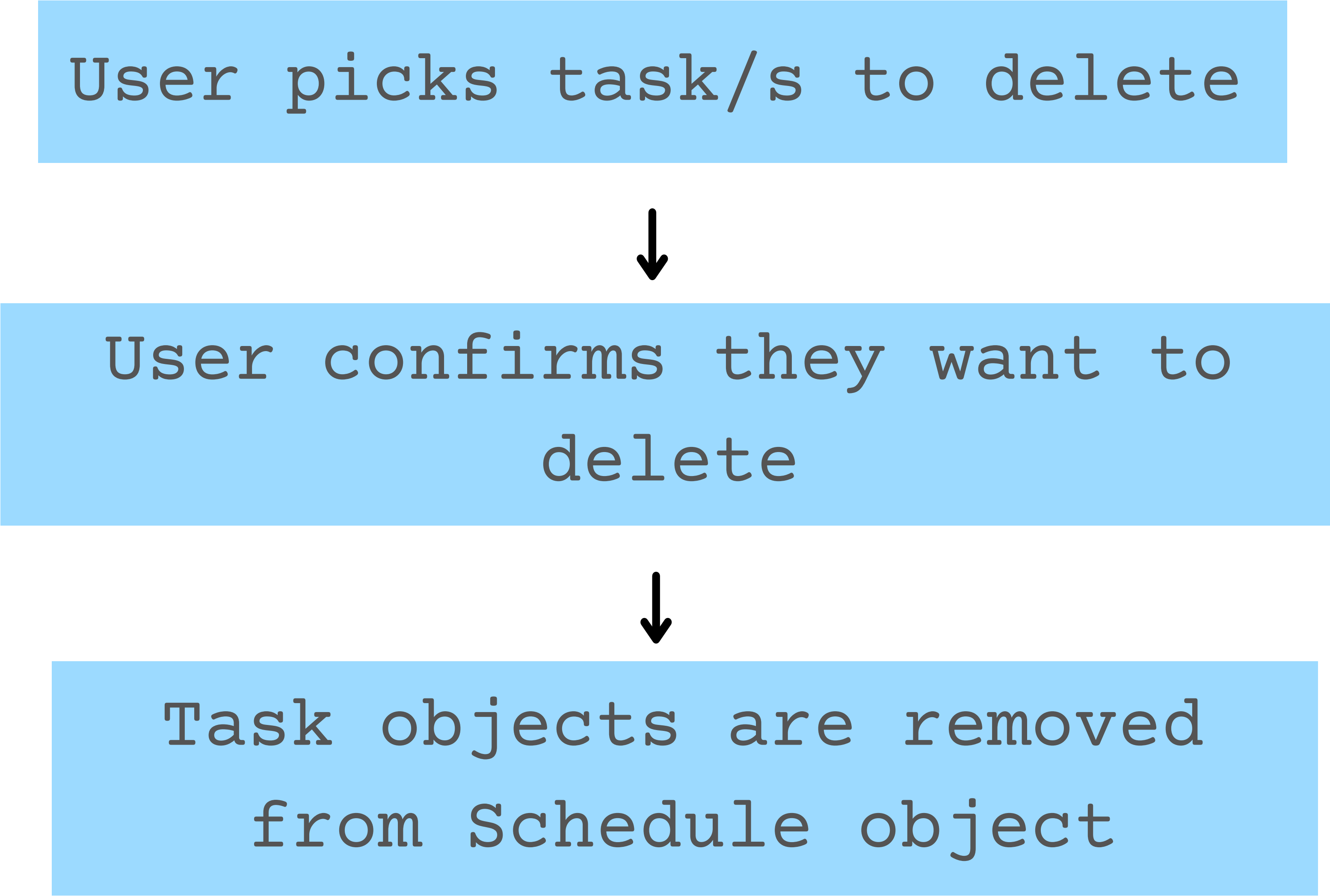
Add task object/s to Schedule object

Techniques used: OOP, loops, error-handling, variables & scope, if-else.

Feature 2: Delete a task

(Demo)

Feature 2: Under the hood



User picks task/s to delete

↓

User confirms they want to delete

↓

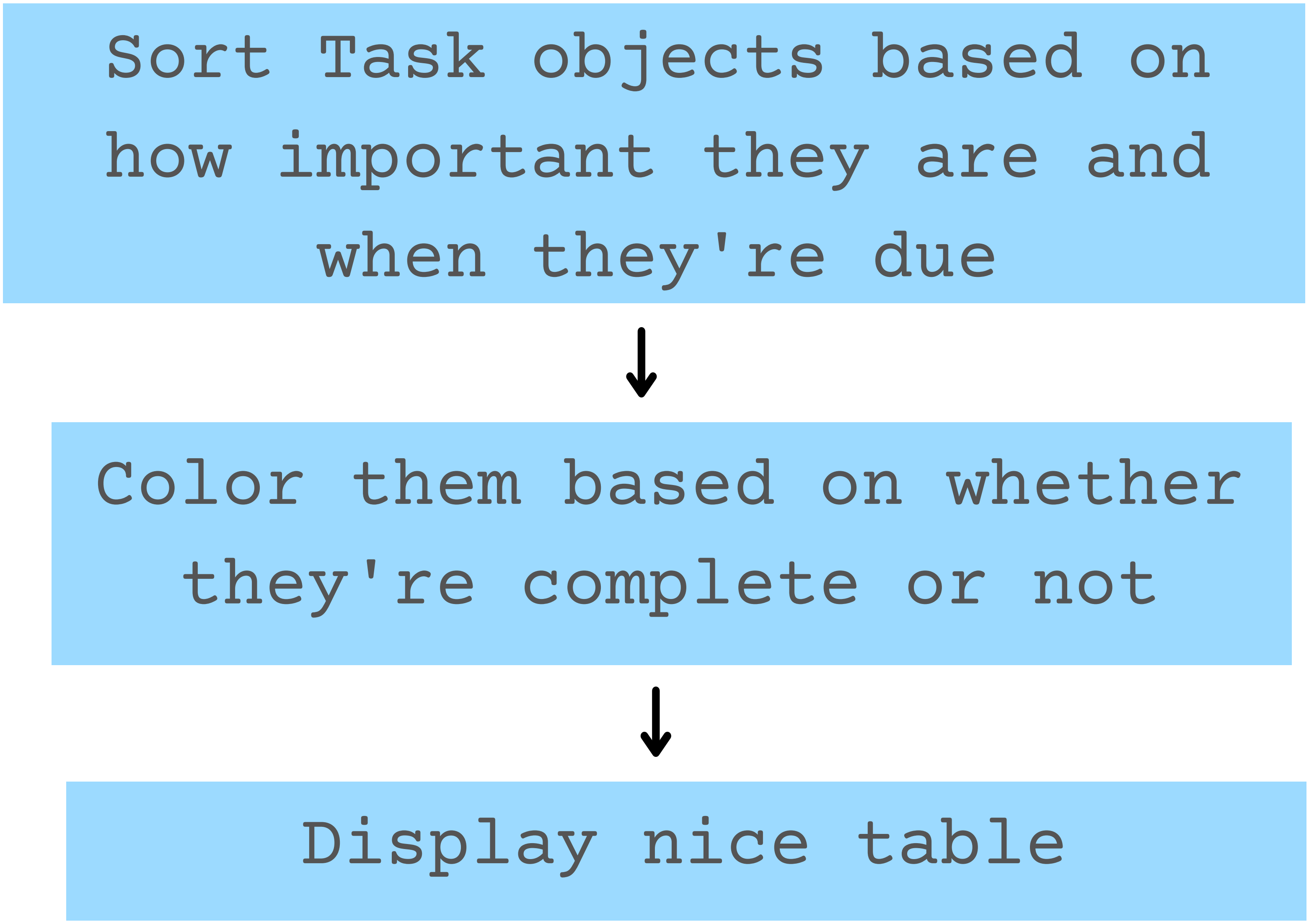
Task objects are removed from Schedule object

Techniques used: OOP, input validation, error-handling, variables & scope

Feature 3: See schedule

(Demo)

Feature 3: Under the hood



Sort Task objects based on how important they are and when they're due

↓

Color them based on whether they're complete or not

↓

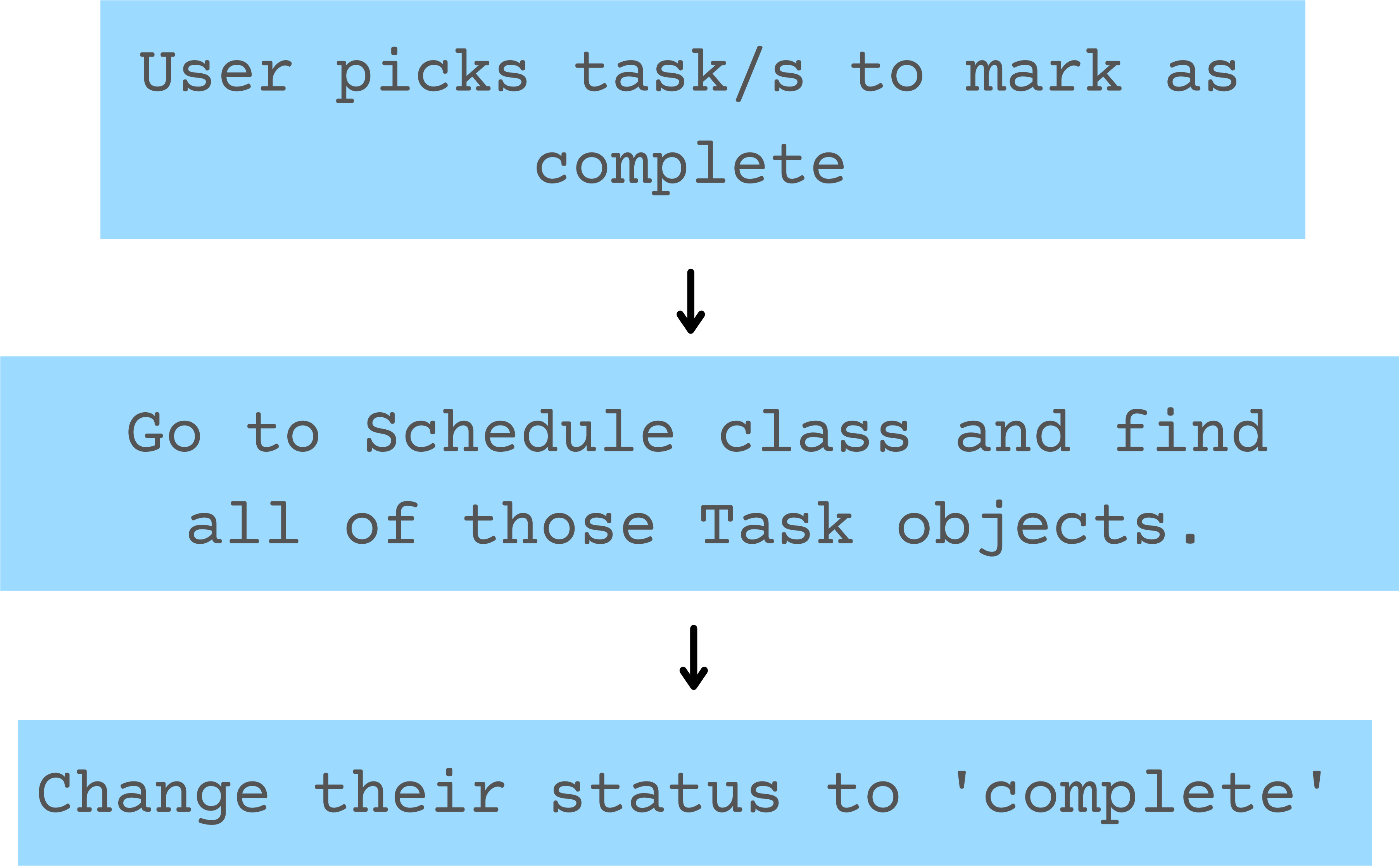
Display nice table

Techniques used: Blocks, iterators, Ruby gems, modules, variables & scope

Feature 4: Mark as complete

(Demo)

Feature 4: Under the hood



User picks task/s to mark as complete

↓

Go to Schedule class and find all of those Task objects.

↓

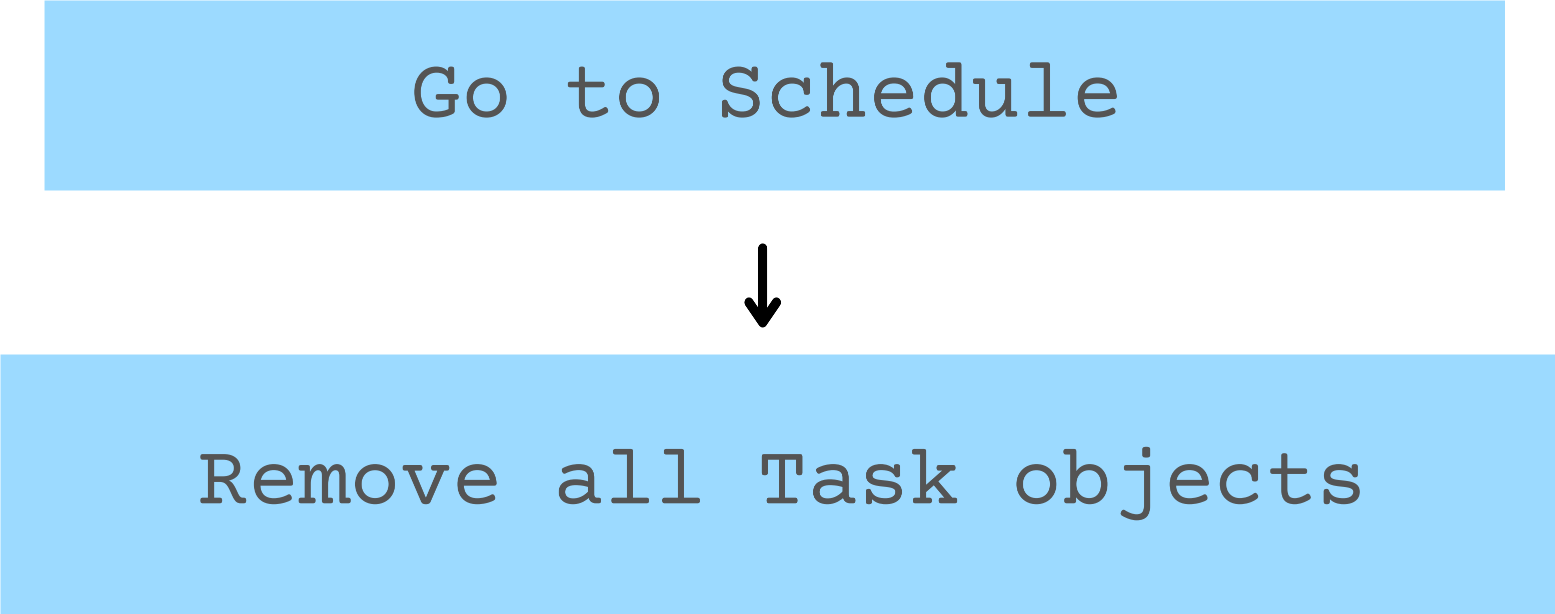
Change their status to 'complete'

Techniques used: OOP, iterators, if-else, variables & scope

Feature 5: Clear schedule

(Demo)

Feature 5: Under the hood



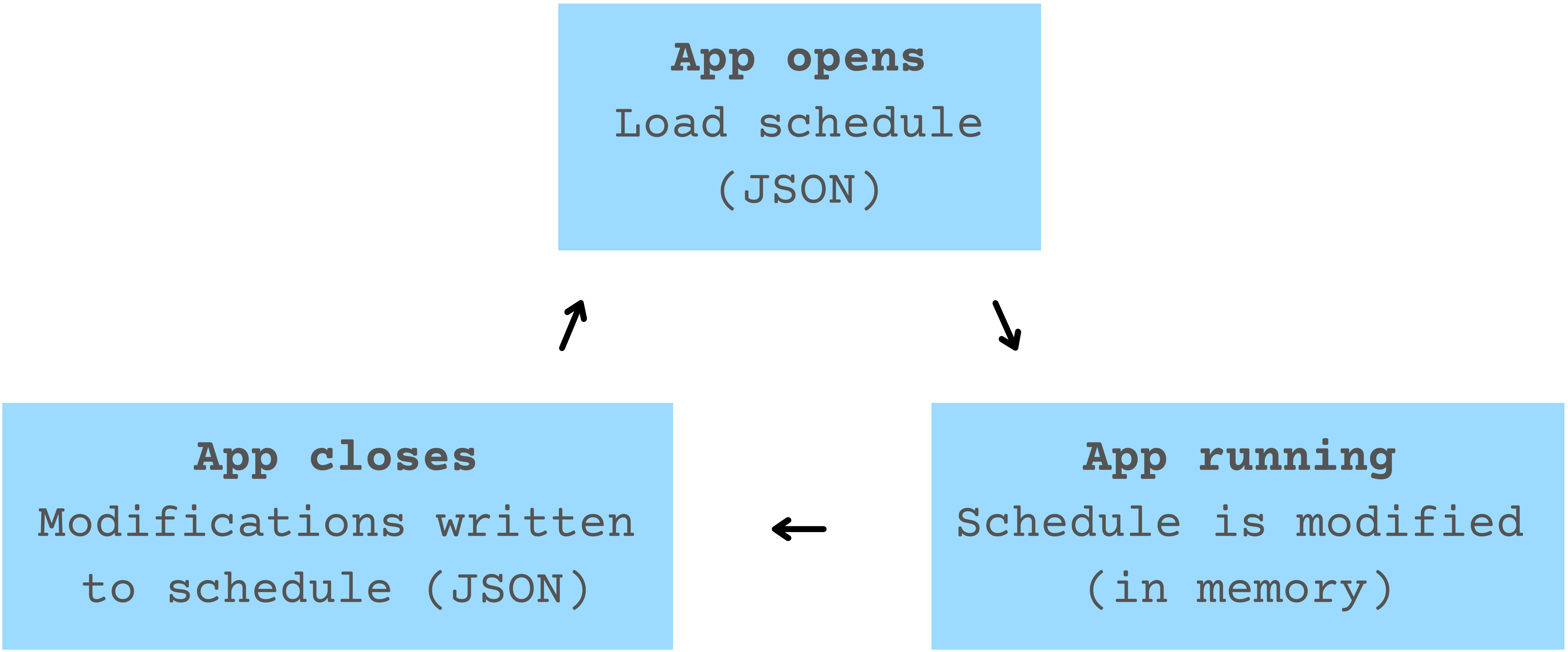
Go to Schedule

↓

Remove all Task objects

Techniques used: OOP, input validation, variables & scope

Note: Persistent storage



App opens
Load schedule
(JSON)

App running
Schedule is modified
(in memory)

App closes
Modifications written
to schedule (JSON)

Techniques used: File-handling, OOP, error-handling, variables & scope

Build process

Iterative planning

TDD

Kanban board

Challenges

Complexity = bugs

DRY code = time

THE END.

Questions?