

# CityRescue Coursework Instructions

*ECM1400 Object-oriented programming*

## Key dates

Partner form deadline	Friday 30 January 2026 (23:59 UK time)
Random pairing (if no form submitted)	Tuesday 3 February 2026
Submission items	ZIP project to ELE + 5-minute video to SharePoint (link file uploaded to ELE)

## 1. What you are doing

In this coursework you will build the back-end for a small emergency dispatch simulator called CityRescue. You will implement the required interface in Java and make your program deterministic so that it can be automatically tested.

You will work in GitHub Classroom, where the starter code is provided and your work is autograded as you push commits.

## 2. Pair programming (required)

This coursework is a pair-programming assignment. You must work with one partner and produce one shared solution.

- Both students should contribute (planning, coding, debugging, testing, documentation).
- You must agree a working pattern (e.g., driver/navigator swaps, or alternating tasks).
- Both students must submit the same final deliverables to ELE.

Working alone is only permitted if you have an ILP exemption.

- If you have an ILP exemption, you must still complete the partner form and indicate that you are working individually.

## 3. Partner form (compulsory)

You must complete the partner form with your partner's name by the deadline below.

[ECM1410 Pair-programming form – Fill out form](#)

- Deadline to submit the form: Friday 30 January 2026 (23:59 UK time).
- If you do not submit the form by the deadline, you will be paired randomly by Tuesday 3 February 2026.
- If you are randomly paired, that pairing stands (unless module staff approve a change).

## 4. Where you will work: GitHub Classroom

The starter code will be provided in GitHub Classroom. You must complete the coursework by committing and pushing your work to your GitHub Classroom repository.

- Clone your repository to your machine (or use GitHub Codespaces if permitted).
- Implement the required classes and methods in Java (do not change the public method signatures).
- Push regularly: your repository will run autograding when you push.
- You will see some autograding feedback and marks, but not the full marking test suite.

## 5. Running and testing locally

This project uses Maven. You can run the tests locally using:

```
mvn test
```

Public tests are included for guidance. Additional hidden tests will be used for the remaining marks.

## 6. What you must submit

### 6.1 ELE submission (ZIP of the project)

When you have finished, download your GitHub repository as a ZIP file and upload it to ELE.

- Your ZIP must include the full Maven project (pom.xml, src/, and any additional files you added).
- Do not submit multiple different versions. Both partners must upload the same ZIP.

### 6.2 5-minute video (SharePoint)

You must record a 5-minute video explaining your solution and upload it to SharePoint.

- Upload the video to SharePoint (make sure staff can access it).
- Create a small file containing the SharePoint link (either a .txt file or a Word document).
- Upload that link file to ELE alongside your ZIP.

### 6.3 Suggested video structure (5 minutes)

- 30–45s: Overview of the problem and your approach.
- 2–3 min: Walkthrough of your main classes and how they interact (OO design).
- 60–90s: Demonstrate key features (dispatching, ticking, obstacles, determinism).
- 30–45s: Testing — show the public tests passing and explain how you tested edge cases.

## 7. Submission checklist

- Both partners must upload the same file.
- Partner form submitted by Friday 30 January 2026 (or ILP exemption indicated).
- All required methods implemented and code compiles.
- Public tests run locally and pass (where applicable).
- Repository pushed to GitHub Classroom.

- ZIP downloaded from GitHub and uploaded to ELE (both partners upload the same ZIP).
- 5-minute video uploaded to SharePoint and accessible to staff.
- SharePoint link file (.txt or .docx) uploaded to ELE (both partners upload the same file).

## 8. Academic integrity and collaboration

You may discuss general ideas with classmates (other than your partner), but your code and final submission must be your own work (yours and your partner's). Do not share solutions or copy code from other pairs.

## 9. Getting help

If you run into problems (GitHub access, pairing, ILP exemption questions, or technical issues), contact [a.chatterjee5@exeter.ac.uk](mailto:a.chatterjee5@exeter.ac.uk) as early as possible. Leave enough time before the deadlines to resolve access issues.