Durham University

Computer Science

Summative Assignment

Module code and title	COMP1101 Programming (Black)
Academic year	2024-25
Coursework title	Collaborative project
Coursework credits	10 credits
% of module's final mark	50%
Lecturer	Steven Bradley
Submission date*	Thursday, 1 May 2025 at 14:00
Estimated hours of work	20
Submission method	Ultra

Additional coursework files	*None*
Required submission items and formats	- Your learning log and your guide for learners should be held within a single private repository on GitHub. - In the repository you should include a README file including a link to the repository on GitHub. - You should submit the entire content of the repository as a zip file

* This is the deadline for all submissions except where an approved extension is in place. For benchtests taking place in practical sessions, the given date is the Monday of the week in which the benchtests will take place.

Late submissions received within 5 working days of the deadline will be capped at 40%.
Late submissions received later than 5 days after the deadline will receive a mark of 0.
It is your responsibility to check that your submission has uploaded successfully and obtain a submission receipt.

Your work must be done by yourself (or your group, if there is an assigned groupwork component) and comply with the university rules about plagiarism and collusion. Students suspected of plagiarism, either of published or unpublished sources, including the work of other students, or of collusion will be dealt with according to University guidelines (https://www.dur.ac.uk/learningandteaching.handbook/6/2/4/).

COMP1101 Programming (Black) Summative Assessment 2

Term 2 Programming Exercise Outline

- Submission by 14:00 Thursday 1 May 2025
- Return after end of exam period
- Contributes 50% of module marks

Subject-specific Knowledge

- A knowledge and understanding of good programming practice (for example, reuse, documentation and style)

Subject-Specific Skills

- an ability to apply reuse by exploiting predefined components
- an ability to use software tools related to programming (programming environments, code management, documentation tools, etc.)
- an ability to apply software development tools and skills in real-world scenarios e.g. open-source projects, hackathons, competitions

Key Skills

- an ability to communicate technical information
- an ability to plan and work independently

Task summary

- Choose a skill to develop
- Choose a collaborative project to contribute to e.g.
  - Open-Source Software
  - hackathon
  - (collaborative) competition
- Record your progress in git with a reflective learning log
- Write a guide for other learners

Skills to develop

- These are examples only
- JavaScript or non-JavaScript

JavaScript Skills

- TypeScript
- React or Vue or Svelte
- Progressive Web Apps
- d3 with SVG
- p5 and openprocessing
- WebGL / WebAudio / Other APIs

Non-JavaScript Skills

- Django
- Continuous Integration (CI/CD)
- Cloud platforms (AWS / Azure / Google / OpenShift / IBM)
- Rust
- Coding with Generative AI

Collaborative project ideas

Contribute to Open Source Software

- Choose your favourite project
- Or pick from a list of things up for grabs for first timers
- Make sure the project is
  - active (frequent and recent commits, not many open PRs)
  - collaborative (multiple contributors)

Don't need to commit code

For example

- Review/ translate docs and help
- Issue gardening (remove duplicates)
- Answer questions on forums e.g. StackOverflow
- Make a video of installation process

Take part in a hackathon

https://www.hackathons.org.uk/events/

Or do your own thing

- Start a new project
- As long as it is collaborative

Requirements and Assessment Criteria

Learning log

- Make at least four separate entries reasonably spaced over a period of at least six weeks
- The entries can either be separate files or different sections within the same file.
- They should be written using Markdown and stored in a private repository on GitHub.
- You need to commit and push each entry as you make it so that the dates of the entries are correctly recorded in git.
- You can include images and external links.

For each entry identify

- What you have done (to learn your skill and contribute to your project)
- What you have learned
- Any changes to your goals (or what your initial goals are for the first entry)
- Next step(s) to achieve your goals

The learning log should have a maximum total word count of 1500 words, as measured by the Microsoft VS Code Word Count plugin and have a reasonable balance between the entries.

Assessment criteria (10% each)

- Number, timing and word count of entries (use of git)
- Appropriate development and monitoring of goals
- Evidence of increased understanding
- Evidence of collaboration
- Evidence of criticality about your own actions and assumptions

Guide for learners

Based on your experience, write a Markdown document which explains to somebody else how to master the skill you have been learning.

You don't need to explain the details, but identify and pull together useful resources.

Guide for learners structure

- Motivation (10%): Why learn it?
- Background (10%): What do you need to know before starting? Include links to material to catch up. Make it clear who your target audience is

- Learning materials (10%): Provide appropriate links to external resources with commentary
- Evaluation (10%): How useful is the skill, compared with the effort of learning it? What similar alternatives are there?
- Presentation (10%)

The presentation mark (10%) will be awarded on the basis of

- formatting in (GitHub flavored) Markdown
- choice of media (images, video)
- clear writing style appropriate for audience

The guide for learners should have a maximum total word count of 1500 words, as measured by the Microsoft VS Code Word Count plugin

Submission

- Your learning log and your guide for learners should be held within a single private repository on GitHub.
- In the repository you should include a README file including a link to the repository on GitHub.
- You should submit the entire content of the repository as a zip file.

Submission repository

- Give me (username stevenaeola) access to the repository on GitHub.
- Allow access to the repository through Settings - Manage Access - Invite a collaborator.
- If you do not allow access you will lose marks.
- Note that this does not affect the access to any code you may write, which would most likely be in a public repository.