TITLE

1 COMP1101 Programming Summative Assessment 1 (DRAFT)

1.1 Term 1 Programming Exercise Outline

- Submission by 14:00 Thursday 17/1/2019
- Return by 14/2/2019
- Contributes 35% of module marks
- Includes peer review feedback which you will be allocated
- Peer reviews need to be submitted by 14:00 31/1/2019
- Quality of your peer reviews contribute 5% to your module mark

---

1.2 Subject-specific Knowledge

- Interaction between JavaScript progams and the Document Object Model (DOM)
- Using control statements to loop and make decisions.
- An understanding of the nature of imperative programming in the object-oriented style.
- A knowledge and understanding of good programming practice (for example, reuse, documentation and style)

1.3 Key Skills

- an ability to recognise and apply the principles of abstraction and modelling

---

1.4 Tasks

- Fork https://github.com/stevenaeola/Durham-p5-lib
- Choose a sketch from openprocessing.org
- Put the original sketch code into a subdirectory of the repository
- Adapt it into a reusable component using JavaScript classes
  - Appropriate constructor
  - Get and set methods for properties
  - `draw` method with optional p5.Renderer as parameter
- Build an example page with properties controlled by form controls
- Write documentation of your code using Markdown

1.5 Submission

- Submit via duo a link to a github (or other git) repository containing your code and documentation
- Make repository public on submission
- Make a pull request to https://github.com/stevenaeola/Durham-p5-lib with your new component

---

1.6 Marking Criteria

Weighted equally

- Usability of code
- Development of original
- Quality of example
- Quality of documentation
- Code quality and management

1.7 Usability of code

- Appropriate parameterisation including defaults
- Encapsulation (private fields where appropriate)
- Useful methods including `draw`

1.8 Development of original

- Original code included in initial commit
- Work done in refactoring code to class
- Work done in useful parameterisation
- Work done in extending scope

1.9 Quality of example

- Need to make an example of your package being used
- HTML page is valid
- Appropriate on-page instructions
- Appropriate on-page controls (form)

1.10 Quality of documentation

- Good ”template” is the documentation for the p5 library itself
- All methods and parameters explained (including constructor)
- Explanation of example
- Source of initial code acknowledged (including licence)

1.11 Code quality: ESLint

Apply rules from eslint.org/docs/rules/:

- Possible Errors
- Best Practices
- Variables
- Stylistic Issues
- ECMAScript 6

1.12 Code management: git

- Appropriate commits including comments
- Consistent development trajectory