

Coursework report 2015–16

CO1112, Creative computing 1

Coursework assignment 1

General remarks

One important purpose of the coursework assignments for Creative Computing 1 is to give students an opportunity to develop their coding skills, in the production of a creative artefact. This was the case in both of these coursework assignments, and in the case of Coursework assignment 1, this was in the context of typography. Again, this is an opportunity for students to learn more about an area related to the main subject material of this course, and to learn how to make connections between topics.

The examiners were pleased to notice that some students had read widely, and had given the topic a great deal of thought. Other students read less widely, but through thought and creativity managed to complete high quality work. The general standard of work submitted was high, with around 25% of students obtaining a mark over 70. There were some weaker submissions too, with around 15% of students failing this coursework assignment.

There were four parts to the coursework assignment. The first three parts involved background work in understanding the topic, and an extended concrete exploration. This contributed half of the marks for the Assignment. The final part, for the other half, required the creation of an artefact, motivated by the work done for the first part. While the division of marks was as mentioned, the effort required does not necessarily follow this division. This is something that it is important for students to understand in a University-level course. Usually, marks are allocated for straightforward work, as well as for more difficult work, and it may be that more effort is needed to do well in a more difficult part. However, this is what demonstrates a student's deeper understanding, and is what the examiners are looking for when awarding higher marks.

Some recurring problems which led to a loss of marks were the following:

- **Follow the submission instructions!** A significant number of students submitted work that did not follow the form required, as specified in the coursework instructions. Students were asked to put the work into a folder containing 2 sub-folders, one called Reports and one called Artefacts; Reports should contain a single pdf, and Artefacts should have 2 folders, one each for Part 2 and Part 3. Many students did not submit their report as a pdf, and others did not stick to the folder structure required. If work is not submitted as specified, students will lose marks needlessly; this will continue in subsequent years, so it is important that students learn to follow these instructions carefully.
- **Submit all subsections asked for!** A significant number of students did not supply work for the different subsections required. Some students simply did not hand in code, while others did not include written explanations. Most significantly, not all students included the critical evaluations asked for in Part 4, thereby losing marks needlessly.

- **Comment your code!** It is essential that you develop good habits right at the start of your career as a coder, and commenting is a highly significant one. Not only does it help the examiner to establish whether you understand the work, and clarifies the approach you have taken to implementation, it is also an invaluable coding tool in itself in that it can help you to identify bugs in your own code. Additionally, if you come back to code after a break, comments help you to remember why you made particular choices.

Part 2 was specifically included to help with this, and though many students were able to incorporate comments into the given sketches, they did not take this forward into either Part 3 or Part 4, showing that they had not really learned much from the exercise of including comments.

Comments on specific questions

Part 1 The requirement here was to write a short essay, discussing two particular aspects of typography, notably kerning, and serif and sans-serif fonts. Many candidates noted that there is a distinction between typeface and font, which the examiners were pleased to see. Students who successfully constructed an essay that included both topics in a coherent way obtained high marks. The tendency is to find material related to the topic and just present summaries of the material found, in no particular coherent order. Students are encouraged to develop an ability to weave together a strong thread through any essay that they write, as this is a skill that will be important in later parts of the degree.

Students were also given an opportunity to practise appropriate academic writing, and were given feedback on this. There were a small number of very nice examples, but there were also unfortunately examples of work with very poor ability to cite reference material.

Essays that attracted high marks showed all of the following: a coherent blending of the two topics of kerning and serif and sans-serif fonts; links to the computerisation aspects; and strong academic writing. For the first, kerning relates to the spacing between letters to obtain the best visual effect, and fonts with and without serifs will have different challenges in this area. For the second, one candidate noted, for example, that in order to automate the manual kerning, software programmers need to describe perception algorithmically, which is a challenging task in itself, and is made more difficult by the fact that perception itself is difficult to define precisely. This is an insightful comment, and was justified by reference to appropriate academic literature. Finally, for strong academic writing, it is important to make use of appropriate material and not rely on Wikipedia or other popular web articles; material must be properly referenced; and appropriate citation (which involved noting in the essay, through connections to the reference list, where the material being presented has been taken from). A list at the end of the essay is not sufficient; it is essential to also include citations within the essay itself.

Any student who at this point is not clear about these aspects of academic writing is strongly urged to find out about this as an integral part of their future study.

Part 2 This part required an exploration of a few software artefacts, and was mostly done in an appropriate way. Exploring the kerning game was usually done well, though the written discussion was sometimes quite superficial. A couple of candidates did insightful work, however, and also linked their discussion with published work on the perception of kerning.

Many students provided reasonable comments of the two *Processing* sketches provided, with a few including very detailed and insightful comments, thereby attracting very high marks. A few students, for example, noted that the `delay(150)` in `text.pde` seemed to provide no functionality, and that the *Processing* documentation does not recommend its use in controlling delay in animations. One student commented that he would replace the line with a `framerate()` method instead.

Some of the code in the given sketches was not of a high quality, and those students who noted this and explained the problems were given additional credit. Developing a critical facility is important, as you proceed through the courses in this degree.

Discussions of the output of the sketches themselves was generally done reasonably well, and included particularly good submissions where the sketches were discussed individually, and then compared with each other.

Part 3 For this, students were to choose one of the two sketches given, and to develop it further. Marks were awarded in three categories: code quality; effectiveness of the extension; and the quality of the discussion. Note that the difference between code quality and effectiveness is in the technical implementation (the coding) compared with the creative idea itself (which relates to effectiveness).

Submissions ranged from fairly unadventurous to more in-depth attempts. Weaker examples may have simply added colour, or attempted to use different fonts, but some of these had a strong discussion which mitigated the creative weakness. Students who explained their motivation generally achieved higher marks; this would usually be found in the report but sometimes students included this information in the code comments as well. One student, for example, explained that they had decided to develop the `text` sketch, as they wanted to experiment with serif and sans-serif fonts, and the impact these can have on what is read. As well as looking at the impact of the message, and the mood created, they also tweaked the code to ensure that all the words were visible and did not run off the screen. Others used the fact that some parts of words were not visible to make a different impact or statement.

Part 4 This final part was worth the most marks, and it was here that examiners expected students to demonstrate both creativity and technical competence. One good way to develop one's creativity is to develop an artefact within specified constraints, and this part was an opportunity for students to gain experience in this kind of activity. Some students rose to the task extremely well, and there were a few excellent submissions, with both original creative work and insightful discussions and critiques.

A notable example was a work that paid tribute to the terror attacks in Paris that had occurred prior to the deadline for submission, that was impactful, moving and technically strong. Another extremely strong work, which was very powerful and also well

discussed, created an image of a Gutenberg press using the text of “Ipsum Lorem”. A ‘user-experience’ artefact based on various images and thoughts inspired by the city of Berlin was also impactful. One of the strongest submissions overall was one that made a strong statement about atrocities against humanity: an animation with the word ‘Why?’ in red is continued with lists of countries where people have suffered in this way is presented randomly across the screen. The creative concept here was very impactful, and the student’s discussion was also very strong, though some more thought about critique (as mentioned below) could have been applied.

Other topics included drink-driving, writing letters, and the creation of small games. A nice idea which could have been taken further was to choose Shakespeare quotes in recognition of the 400th anniversary of Shakespeare’s death. An attempt at including work from Shel Silverstein and linking it visually with emotion was a very nice idea, which the student did not manage to implement fully. However, they did explain what they had hoped to do, and also discussed the work they had succeeded in completing, and so got a reasonable mark. It’s always important to explain what you’ve done when this is appropriate.

Many students explained their design decisions, which was asked for, but not all gave a critique (evaluation) of their work, which was also asked for. It is essential to include both. One way to see this is that the design decisions describe what you did and why; and the critique describes how well they worked, how effective they were in achieving what you had planned. You should always approach your creative computing work with these kinds of questions in mind, and you should also get into the habit of looking at the work of others with these questions in mind, to help you get a sense of what works, what has impact, and what fails – and why. Some students also included the ergonomic aspects (which relate to how easy something is to use) and some considered aesthetics in their critique, which is also a good thing to do.

Overall, this coursework assignment was approached well; most students completed all four parts to a good extent, and there were a handful of excellent submissions where all sections were done very well, resulting in almost full marks.