

COM1003 Java Programming

Autumn Semester 2020-21

Programming Assignment 2

Dr Siobhán North
Department of Computer Science, The University of Sheffield

Learning outcomes

This assignment will assess your ability to:

- Write a program from a specification;
- Write clear, good quality program code;
- Use loops in Java;
- Use the `sheffield` package for graphical output;
- Use arrays in Java.
- Use the `Math` class in Java

It is worth 20% of your mark for the first semester of the module and must be submitted by 30 November 2020. You will find information about the exact deadline, the marking scheme and how you must submit your work at the end of this document.

The specification below is quite precise in telling you what sort of input you should expect and what you must output but usually does not tell you how to get from one to the other. This is deliberate and part of the test. However it is not meant to be ambiguous so if there is something you don't understand you can ask questions, ideally by email. I will put all the email questions and their answers on an FAQ page as they come in. Even if you think you understand everything it is a good idea to check this page before you submit.

You do not need to do everything described below to hand in your work and get marks for it. The marks will be awarded depending on how much you have achieved (see below for details) but if you hand in a program it must compile and run to get any marks. So a well written program that does something is always better than a program which would do more if it had worked.

The Program

You must write a program to read in information from a file and use it to display a picture. As in the earlier assignment, there are various increasingly difficult versions of this task each with an associated maximum number of marks that can be obtained. I suggest you attempt the easiest one, get it to work and back

up your program (and possibly upload it to Blackboard) before you attempt the next one. For all versions your program must be called `Dino.java`; I will be able to recognise which version you are attempting from the output. You should only hand in *one* version, obviously the hardest one you can make work, not all the easier versions as well. This is part of the specification.

Stage 1 - A Dinosaur

You have been provided with a file called `dino.txt` containing 4,305 lower case characters with no line breaks. These can be decoded to make a picture of a dinosaur. The first character in the file corresponds to the top left pixel in the picture of the dinosaur and the last is the bottom right. To make the picture they need to be displayed as 35 rows each of which are 123 pixels wide. The decoding is easy; if the character forms part of the dinosaur its Unicode value is even and if it doesn't the value is odd.

Read in the dinosaur file, store the information from it in an appropriate array and display it in an `EasyGraphics` screen of the same dimensions as the picture. You can choose the colours yourself but make sure there is sufficient contrast between the dinosaur to make the picture clear.

It is possible to do this without using array but you will need the array for later parts of the assignment and, if you don't use an array, you will not get the marks associated with use of arrays.

If you do this with a perfectly written program you can expect to get 30%



Stage 2 - A Bigger Dinosaur



Modify your earlier program to scale both the screen and the picture up four times. So the window should be four times as high and four times as wide and the picture should be scaled up in proportion.

If you do this perfectly you can expect to get 50%. If you submit this version you should *not* submit the 30% version.

Stage 3 - Another Dinosaur

Modify your previous solution to double the width of your graphics window and display two copies of the dinosaur. Include an horizon as well as shown in the picture to the right.



If you do this perfectly you can expect to get 65%. If you submit this version you should *not* submit the 30% or 50% versions.

Stage 4 - A Third Dinosaur



Modify your earlier program to add a third copy of the dinosaur in the centre of the graphics window, horizontally.

If you do this perfectly you can expect to get 80%. If you submit this version you

should *not* submit the 30%, 50% or 65% versions.

Stage 5 - With a Stary Sky

Modify your earlier program to add stars to the sky in the background. The sky should be very dark and the stars white or yellow and the colour of the dinosaur should show up against both.



There must be 50 stars randomly placed in the sky. They may not all show up because some are likely to be behind one or other of the dinosaurs. By random I mean they must be placed at random on each run of the program so different runs will have different patterns of stars.

If you do this perfectly you can expect to get 100%. If you submit this version you should *not* submit the 30%, 50%, 65% or 80% versions.

Submission and deadline

You must submit a file called Dino.java via the submission point on Blackboard by 3pm on Monday 30th November. Do not submit anything else and do not submit it in any other format. Blackboard is very bad at displaying Java programs so you may think that Blackboard has messed up the layout of your program but I will download the program text and the layout will be preserved.

Late work will be penalised using the standard University scale (a penalty of 5% per working day late; work will be awarded a mark of zero if it is more than 5 working days late).

This is an individual assignment. You must work on it alone and hand in your own work. If you work collaboratively and then pretend you did the work alone we will find out (we have a very good plagiarism checker and all submitted work will go through it) and, as you have already been told or are about to be told during your first group tutorial, we take the use of unfair means in the assignment process very seriously. Don't even think about handing in work you didn't do yourself.

The Marking Scheme

The mark for this assignment is worth 20% of the first semester mark and so 10% of the overall mark for COM1003.

The criteria for program style quality are the same as for the first assignment so do not submit this assignment until you have checked the feedback for the first assignment. That will be released by 16th November.

The marking scheme for the various versions is as follows:

Version	1	2	3	4	5
Use of arrays	7	13	16	19	20
Use of loops	7	7	9	11	13
Use of Graphics window	6	13	18	23	25
Use of Random numbers					8
Style	10	17	22	27	34
Total	30	50	65	80	100

The style mark will be calculated as a percentage as follows and scaled to

Readable layout	18
Correct use of identifiers	18
Correct use of types	18
Useful comments	18
Use of constants	18
Meets the specification	10
Total	100

the maximum for whatever version you submit.