

# Fundamentals of Programming

---

## Assignment Option 1: PaSciRo Simulation

Semester 2, 2020 v1.0

**Discipline of Computing**  
**Curtin University**

### 1 Preamble

In practicals you have implemented and learned about simulations, object-orientation and (soon) how to automate the running of multiple simulations. In this assignment, you will be making use of this knowledge to extend a given simulation to provide more functionality, complexity and allow automation. You will then report on the results generated by the simulation.

### 2 The Problem

We will be simulating life on the planet PaSciRo, which is inhabited by three lifeforms of interest, each with their own behaviour and interaction rules. You will be given some sample code, simulating the population using arrays. For the assignment, you will convert the code to model the planet creatures using objects, and to add features to the simulation (e.g. more functionality, statistics, graphics, parameter sweeps). Your task is to extend the code and then conduct an experiment, varying the input parameters, to see how they impact the overall simulation.

The required extensions are:

1. **Object Behaviour:** Extend to have the lifeforms represented as objects. Each type of object will have its own behaviour(s). This should be implemented via the object's methods and attributes.
2. **Visualisation:** Enhance the output to vary the representation of animals by their type, and possibly direction of movement. You may choose an "image" representation, to improve on using coloured dots. You should also be able to save the simulation state as an image or a csv file.
3. **Movement:** Movement should vary by lifeform and needs to be constrained to the space. The Base code movement is highly unrealistic.  
**Prompts:** Consider how they might move, e.g. Do the different lifeforms move in different ways? Do they keep going in the same direction unless they have a "change direction" event at some level of probability.
4. **Collisions:** The animals are moving in a two dimensional space. Your code should recognise when collisions occur and respond accordingly.  
**Prompts:** How do you recognise a collision? Add rules as to what happens when different combinations of lifeforms collide. Your simulation should also include mechanisms for reproduction and death within the populations – probably related to certain collisions.
5. **Boundaries:** Enhance the rectangular grid with boundaries to restrict movement.

**Note:** Your program should allow command line arguments to control the parameters of the experiment/simulation.

Your code should include comments to explain what each section does and how. It is useful to keep track of your changes in the comments at the top of the program. Feel free to re-use the code and approaches from the lectures and practicals. However, remember to cite and self-cite your sources. If you submit work that you have already submitted for a previous assessment (in this unit or any other) you have to specifically state this.

Beyond the working program, you will submit two documents: the Simulation Project Documentation and a Simulation Project Report on your experiment. There will be **bonus marks** for additional functionality and the use of more advanced programming techniques (e.g. interactivity, high quality visualisation, 3D space, parameter sweep etc.) but only if they're sensible and done well. Make sure to discuss the additional work in your Documentation and Report, this will be easy if you make notes and keep old (incremental) versions of your code.

## Remember : Think before you code!

### 3 Submission

Submit electronically via Blackboard. Make sure to submit early. You can submit multiple times – we will only mark the last attempt. Take care not to submit your last version late though. Read the submission instructions very carefully.

You should submit a single file, which should be zipped (.zip). Check that you can decompress it on lab machines. These are also the computers that your work will be tested on, so make sure that your work runs there. The file must be named FOP\_Assignment\_<id> where the <id> is replaced by your student id. There should be no spaces in the file name; use underscores as shown.

The file must contain the following:

- Your **code**. This means all files needed to run your program. That includes input files used as part of the assignment if that is required to run your program.
- **README** file including short descriptions of all files and dependencies, and information on how to run the program.
- **User Guide** and **Report** for your code, as described in Section 3.1.
- A signed and dated **cover sheet**. These are available on Blackboard. You can sign a hard copy and scan it in or you can fill in a soft copy and digitally sign it.

Make sure that your zip file contains what is required. Anything not included in your submission will not be marked, even if you attempt to provide it later. It is your responsibility to make sure that your submission is complete and correct.

#### 3.1 Documentation and Report

You need to submit your Documentation and Report in docx, pdf or Jupyter Notebook format. The **Simulation Project Documentation** is where you describe how you approached the implementation of the simulation, and explain to users how to run the program. The **Simulation Project Report** takes the application(s) you have developed, and uses them to explore the simulation outputs. This exploration would include changing parameters, simulation time and perhaps comparing outcomes if you switch various features on/off.

Your **Documentation** will be minimum 2-4 pages and should include the following:

- An **overview** of your program's purpose and features.
- A guide on how to use your simulation and parameter sweep code, if applicable.
- A discussion of your code, explaining how it works, any additional features and how you implemented them.

The **Report** will follow the structure of a standard academic report or paper. It should be at least 4-6 pages long. Required sections are:

- **Abstract:** Explain the purpose of the report and state the parameters you have investigated, and the outcomes/recommendations.
- **Background:** Discuss the purpose of the simulation and your choice of parameters.
- **Methodology:** Describe how you have chosen to set up and compare the simulations, and why. Include commands, input files, outputs – anything needed to reproduce your results.
- **Results:** Present the results of your simulations.
- **Conclusion and Future Work:** Give conclusions and what further investigations could follow.

### 3.2 Marking

Marks will be awarded to your submission as follows:

- **[30 marks] Code Features.** Based on your implementation and documentation
- **[30 marks] Demonstration.** Students will demonstrate their code and respond to questions from the markers.
- **[10 marks] Documentation.** As described in section 3.1.
- **[30 marks] Report.** As described in section 3.1, with the majority of marks for the methodology and results sections.

Marks will be deducted for not following specifications outlined in this document, which includes incorrect submission format and content. If the cover sheet isn't provided with your submission, your submission will not be marked and you will be awarded zero (0) marks. If you forget to submit the cover sheet you will be allowed to submit it separately to the unit coordinator (by e-mail or in person) but will lose 5 marks.

### 3.3 Requirements for passing the unit

Please note: As specified in the unit outline, it is necessary to have attempted the assignment in order to pass the unit. As a guide, you should score at least 15% to be considered to have attempted this assignment. We have given you the exact mark breakdown in Section 3.2. Note that the marks indicated in this section represent maximums, achieved only if you completely satisfy the requirements of the relevant section.

Plagiarism is a serious offence. This assignment has many correct solutions so plagiarism will be easy for us to detect (and we will). For information about plagiarism, please refer to <http://academicintegrity.curtin.edu.au>.

In the case of doubt, you may be asked to explain your code and the reason for choices that you have made as part of coding to the unit coordinator. A failure to adequately display knowledge required to have produced the code will most likely result in being formally accused of cheating.

Finally, be sure to secure your code. If someone else gets access to your code for any reason (including because you left it on a lab machine, lost a USB drive containing the code or put it on a public repository) you will be held partially responsible for any plagiarism that results.

### **3.4 Late Submission**

As specified in the unit outline, you must submit the assignment on the due date. Acceptance of late submissions is not automatic and will require supporting documentation proving that the late submission was due to unexpected factors outside your control. See the unit outline for details as to the procedure for requesting that an assessment be accepted after the due date.

Also note that IT related issues are almost never a valid excuse.

In the event that you submit your assignment late and are deemed to have a valid excuse, you will be penalised 10% (that is, 10% out of 100%, not out of what you would have received) per calendar day that you are late, up to a maximum of seven (7) calendar days. Any work submitted after this time will not be marked and you will automatically fail the unit. Note that if you are granted an extension you will be able to submit your work up to the extended time without penalty – this is different from submitting late.

### **3.5 Clarifications and Amendments**

This assignment specification may be clarified and/or amended at any time. Such clarifications and amendments will be announced in the lecture and on the unit's Blackboard page. These clarifications and amendments form part of the assignment specification and may include things that affect mark allocations or specific tasks. It is your responsibility to be aware of these, either by attending the lectures, watching the iLecture and/or monitoring the Blackboard page.